



Escola d'Enginyeria de Telecomunicació i  
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL FINAL DE GRAU

**TÍTOL DEL TFG:** Sistema de navegació indoor de baix cost per drons

**TITULACIÓ:** Grau en Enginyeria de Sistemes Aeroespacials

**AUTOR:** Sergi Lucas Millan

**DIRECTOR:** Oscar Casas Piedrafita

**DATA:** 24 de juliol de 2019

**Títol:** Sistema de navegació indoor de baix cost per drons

**Autor:** Sergi Lucas Millan

**Director:** Oscar Casas Piedrafita

**Data:** 24 de juliol de 2019

## Resum

El treball de fi de grau està basat en el disseny d'un sistema de navegació per a espais interiors de baix cost que sigui capaç de fusionar-se amb un dron. Per tal de provar el sistema que dissenyem es construeix un dron per fusionar-lo al sistema i fer les proves de vol. El treball està estructurat en quatre grans capítols.

El primer capítol estudia l'estat de l'art actual sobre els diferents sistemes de navegació que existeixen i s'estan utilitzant. Es fa un estudi de cadascun de les existents possibilitats. Com a resultat d'aquest estudi, la tecnologia escollida per l'ús en el sistema de navegació és l'òptica, és a dir, la basada en informació provinent de càmeres. Les càmeres escollides són la càmera de profunditat Intel RealSense D435 i la càmera de seguiment Intel RealSense T265.

El segon capítol resumeix les lleis i equacions matemàtiques amb les quals modelem matemàticament els moviments d'un dron. També tracta com funciona el controlador PID que fan servir la gran majoria dels controladors de vol dels drons. Aquest capítol serveix per entendre com volen els drons i com s'estabilitzen.

En el tercer capítol expliquem l'elecció de tots els components amb els quals hem construït el dron per tal de poder posar a prova el sistema de navegació indoor. Un dels principals criteris a l'hora de seleccionar els elements del dron ha sigut el cost d'aquests. Finalment fem dues proves de vol per entendre la diferència entre volar sota el senyal GPS i volar en un espai interior on no arriba el senyal GPS.

L'últim capítol parla del disseny del sistema de navegació indoor. Un cop el sistema de navegació indoor ha estat dissenyat i provat amb el dron, els resultats obtinguts han sigut positius. Primer de tot hem comprovat la nul·la capacitat que té un dron de navegar en un espai interior sense senyal GPS. I posteriorment hem vist que amb la utilització del sistema de navegació indoor ha sigut possible navegar en un espai on no arriba el senyal GPS.

**Title:** Low cost indoor navigation system for drones

**Author:** Sergi Lucas Millan

**Director:** Oscar Casas Piedrafita

**Date:** July 24, 2019

## Overview

The final degree project is based on the design of a low cost navigation system for indoor environments that is able to use in a drone. To prove the system in a real experiment a drone is built to use with the designed navigation system and to make flight tests. The project is structured in four big chapters.

The first chapter studies the actual state of art of the different navigation systems that exists and are used. We make a study of each of the existing possibilities. As a result of the study, the selected technology to use in the navigation system is the optic, that is to say, the data comes from cameras. The selected cameras are the depth camera Intel RealSense D435 and the tracking camera Intel RealSense T265.

The second chapter summarizes the laws and mathematical equations with which we mathematically model drone movements. It also explains how PID controllers work. This controller is the most commonly used controller in drone flight controllers. This chapter serves to understand how drones fly and how they stabilize.

In the third chapter, we expose all the selected components used to build the drone in order to test the indoor navigation system. One of the main criteria to select the drone components was the cost of them. Finally, we made two flight test to understand the difference between fly in an outdoor space with GPS signal and fly in an indoor space without GPS signal.

In last chapter, we explain the design of the indoor navigation system. Once the indoor navigation system was built and test with the drone, the results were very satisfactory. First of all we prove the zero capacity that a drone has to navigate in an indoor space without GPS signal and then, we have observed that using the indoor navigation system designed in this project, the drone is able to navigate in any environment without GPS signal.

# ÍNDEX

<b>INTRODUCCIÓ .....</b>	<b>1</b>
<b>CAPÍTOL 1. ESTAT DE L'ART .....</b>	<b>2</b>
<b>1.1. Mètodes posicionament.....</b>	<b>2</b>
1.1.1. Mètodes basats en la triangulació .....	3
1.1.2. Mètodes basats en la potència de senyal .....	4
1.1.3. Fingerprinting.....	4
1.1.4. Cell of Origin (CoO) .....	5
1.1.5. Navegació per estima .....	5
1.1.6. Filtre de Kalman .....	5
<b>1.2. Tecnologies posicionament .....</b>	<b>6</b>
1.2.1. Wifi.....	6
1.2.2. Bluetooth Low Energy .....	6
1.2.3. ZigBee .....	7
1.2.4. Ultra Wide Band .....	8
1.2.5. RFID .....	9
1.2.6. Infrarojos.....	10
1.2.7. Ultrasons.....	10
1.2.8. Basat en imatges.....	11
1.2.9. Basat en la inèrcia .....	12
1.2.10. Lidar.....	13
<b>1.3. Elecció tecnologia .....</b>	<b>14</b>
<b>CAPÍTOL 2. MODEL MATEMÀTIC I CONTROL .....</b>	<b>16</b>
<b>2.1. Sistemes de coordenades .....</b>	<b>16</b>
<b>2.2. Equacions dinàmiques .....</b>	<b>18</b>
2.2.1. Forces.....	18
2.2.2. Moments .....	20
<b>2.3. Sistema de control .....</b>	<b>22</b>
<b>CAPÍTOL 3. DISSENY DRON .....</b>	<b>25</b>
<b>3.1. Elecció dels components .....</b>	<b>25</b>
3.1.1. Estructura .....	25
3.1.2. Controlador de vol .....	26
3.1.3. Motors.....	27
3.1.4. Controlador de velocitat.....	28
3.1.5. Bateria .....	28
<b>3.2. Resum dron.....</b>	<b>29</b>
<b>3.3. Primer vol .....</b>	<b>30</b>
<b>CAPÍTOL 4. DISSENY SISTEMA DE NAVEGACIÓ .....</b>	<b>33</b>
<b>4.1. Eines hardware .....</b>	<b>33</b>

4.1.1.	Intel RealSense .....	33
4.1.2.	Ordinador de bord .....	35
<b>4.2.</b>	<b>Eines software .....</b>	<b>36</b>
4.2.1.	Llenguatge de programació .....	36
4.2.2.	Comunicació amb les càmeres .....	36
4.2.3.	Comunicació amb el pilot automàtic .....	37
<b>4.3.</b>	<b>Funcionament del sistema de navegació .....</b>	<b>38</b>
4.3.1.	Obtenció dades provinents de les càmeres .....	39
4.3.2.	Enviament dades al controlador de vol .....	42
4.3.3.	Funcionament i lògica del sistema .....	44
<b>4.4.</b>	<b>Resultats .....</b>	<b>45</b>
4.4.1.	Prova de terra .....	46
4.4.2.	Prova de posicionament .....	48
4.4.3.	Prova de navegació .....	49
<b>CONCLUSIONS .....</b>		<b>50</b>
<b>BIBLIOGRAFIA .....</b>		<b>52</b>
<b>ANNEXOS .....</b>		<b>57</b>
<b>1.</b>	<b>Muntatge dron .....</b>	<b>57</b>
<b>2.</b>	<b>Configuració dron .....</b>	<b>61</b>
<b>3.</b>	<b>Vídeos proves de vol .....</b>	<b>63</b>
3.1.	Amb senyal GPS .....	63
3.2.	Sense senyal GPS .....	63
3.3.	Comprovació sistema de navegació indoor .....	64
3.4.	Navegació amb el sistema de navegació indoor .....	64
<b>4.</b>	<b>Codi .....</b>	<b>64</b>
4.1.	Main .....	64
4.2.	Funcions relacionades amb les càmeres .....	66
4.3.	Funcions relacionades amb el dron .....	68

# ÍNDIX DE FIGURES

Fig. 2.1 Estructura d'un quadricòpter.....	16
Fig. 2.2 Estructura d'un controlador PID.....	23
Fig. 3.1 Imatge del dron amb les peces seleccionades.....	30
Fig. 3.2 Imatge del dron durant la realització del vol amb senyal GPS en un espai exterior .....	31
Fig. 3.3 Imatge del dron durant el vol en un espai interior sense senyal GPS .	31
Fig. 4.1 Connexió entre la Raspberry Pi i la Pixhawk 2.4.8 a través dels ports UART .....	35
Fig. 4.2 Estructura paquet MAVLink.....	37
Fig. 4.3 Sis graus de llibertat de la càmera T265 .....	39
Fig. 4.4 Sistema de referència Intel RealSense T265 .....	40
Fig. 4.5 Sistema de referència de la càmera T265 comparat amb sistema de referència del dron .....	41
Fig. 4.6 Visualització de la trajectòria del dron durant la prova en terra vist des de l'estació de control de terra (Mission Planner) .....	46
Fig. 4.7 Trajectòria del dron durant la prova de terra provinent de les dades emmagatzemades de la càmera de seguiment.....	47
Fig. 4.8 Inspector de missatges MAVLink durant la prova de terra .....	47
Fig. 4.9 Trajectòria del dron durant la prova de vol per comprovar el posicionament del dron amb el sistema de navegació dissenyat.....	48
Fig. 4.10 Trajectòria del dron durant la prova de vol per comprovar navegabilitat amb el sistema de navegació dissenyat.....	49
Fig. Annexos 1 Soldadura variadors a la placa .....	58
Fig. Annexos 2 Gir dels motors .....	58
Fig. Annexos 3 Controlador de vol Pixhawk.....	59
Fig. Annexos 4 Connexions dels motors .....	60
Fig. Annexos 5 Dron construït .....	61

## INTRODUCCIÓ

Un VANT (vehicle aeri no tripulat), dron o UAV (Unmanned Aerial Vehicle) és un tipus d'aeronau, reutilitzable i sense persones a bord que pot estar controlat totalment per un humà o pot tenir un cert grau d'autonomia, i es fan servir tant en tasques de caràcter civil com militar.

En els últims anys l'ús dels drons ha crescut de forma exponencial. La gran consultora multinacional PwC (PriceWaterhouseCoopers) (veure [1]) estima que els drons mouen 127.000 milions d'euros arreu del món . Això és degut a la gran versatilitat que tenen els drons i la capacitat de poder operar en zones de difícil accés pels humans, els vehicles terrestres i els vehicles aeris tripulats.

Actualment, el món dels drons a Espanya està en plena evolució. Està previst que el sector serà un dels que major creixement viurà en un termini curt-mitjà. Es preveu que en tan sols 15 anys el volum de negoci es multipliqui per 30 (veure [2]). Per efectuar aquesta evolució el Ministeri de Foment del Govern d'Espanya ha elaborat el "Plan Estrategico para el desarrollo civil de los drones en España" (veure [3]) pel període que compren entre 2018 i 2021. Aquest pla assegura que el sector del dron sigui un negoci segur.

L'evolució dels drons al món està esperant canvis en les regulacions de la seguretat aèria. L'empresa europea SESAR JU ha creat la proposta *U-Space*. *U-Space* és un projecte per la Unió Europea per a la integració dels drons en tot l'espai aeri d'Europa (veure [4]).

A hores d'ara la gran majoria de drons utilitzen un sistema de posicionament GPS, aquest sistema proporciona una posició del dron per a poder mantenir la posició d'aquest. El problema del GPS és que només funciona en exteriors, per tant l'ús dels drons en interiors és limitat i la majoria d'aplicacions actuals per drons es desenvolupen a l'exterior.

La navegació és l'art de conduir un vehicle d'un lloc a un altre, de forma eficient i amb responsabilitat. La navegació d'un dron és possible gràcies al sistema GPS que porta incorporat. Com acabem de veure un dels problemes del GPS és que no arriba a espais interiors, per tant, la navegació dels drons en interiors no es pot basar en el GPS.

Doncs, la motivació del treball és el desenvolupament d'un sistema de navegació per interiors i la fusió d'aquest sistema amb un dron. L'objectiu del treball és poder desenvolupar un sistema de navegació indoor que sigui precís i de baix cost. Un altre objectiu és la configuració i construcció des de 0 d'un dron en el qual puguem provar el sistema de navegació indoor.

## CAPÍTOL 1. ESTAT DE L'ART

La navegació és l'art i la ciència de conduir un vehicle d'un lloc a un altre de forma eficient i amb responsabilitat. La navegació prové de la prehistòria, els primers humans utilitzaven els accidents geogràfics que es trobaven al seu voltant com a punts de referència per poder trobar el seu camí en àrees grans de mar obert. Actualment el sistema de posicionament que utilitzem com a eina per a la navegació és el GPS (*Global Positioning System*). El GPS és un sistema que permet determinar la posició en qualsevol punt de la Terra. És un sistema que va ser desenvolupat pel Departament de Defensa dels EUA. El GPS està format per una constel·lació de mínim 24 satèl·lits que donen cobertura a tota la Terra. La tècnica que utilitza per calcular la posició es coneix com a trilateració, i necessita almenys 4 satèl·lits per funcionar. La precisió del GPS per a ús civil és d'uns pocs metres. Un dels grans inconvenients del GPS és que no funciona dins dels edificis. Això és degut a l'atenuació provocada pels materials amb què s'han construït l'edifici i també amb l'efecte multicamí que produeix el rebot del senyal dins de l'edifici. Per aquest motiu el sistema més utilitzat per a la navegació habitual no es pot utilitzar per a la navegació en interiors.

Avui dia la navegació en interiors està despertant un gran interès per a la investigació donat la gran utilització dels dispositius mòbils en el dia a dia. Les aplicacions que es beneficien d'aquest tipus de navegació es troben en magatzems, pàrquings soterrats, centres comercials, etc. També és degut a la robòtica i tot el món que l'envolta, ja que és una branca de la tecnologia on s'està investigant més degut a les grans aplicacions que sorgeixen.

Com hem mencionat en la introducció, l'objectiu del treball és desenvolupar un sistema de navegació en interiors que sigui precís i de baix cost. Per poder fer la tria de la tecnologia que utilitzarem, primer hem d'investigar l'estat de l'art del posicionament indoor, és a dir, quines són les millors tecnologies per aquestes aplicacions. A més, la majoria d'aquestes aplicacions funcionen a través d'ones, i acostumen a utilitzar els mateixos mètodes per calcular la posició. Per tant, primer estudiarem els diferents mètodes.

### 1.1. Mètodes posicionament

Independentment de la tecnologia que utilitzem per a calcular la posició d'un objecte, tot el posicionament necessita uns mètodes i mesures procedents de diverses fonts. Un cop hem obtingut aquestes mesures provinents de les fonts, segons el tipus de mesures obtingudes podrem calcular la posició utilitzant diferents mètodes (veure [5],[6],[7],[8],[9],[10]).



### 1.1.1. Mètodes basats en la triangulació

La triangulació és un mètode que utilitza la geometria per a determinar posicions de punts, mesures de distància o àrees de figures. Per a posicionament indoor existeixen diferents mètodes basats en la triangulació, que depenen de la informació que utilitzem.

#### 1.1.1.1. Temps d'arribada (ToA) o Temps de vol (ToF)

Aquesta tècnica es basa en el fet que un senyal tarda un temps a arribar entre el node emissor i el node receptor. Si sabem la velocitat de transmissió del senyal i som capaços de mesurar el temps d'arribada podem calcular la distància. La fórmula per calcular la distància  $d$  sent  $v$  la velocitat de propagació del senyal i  $T$  el temps de propagació:

$$d = T \times v \quad (1.1.)$$

Per utilitzar aquesta tècnica serà necessari la utilització de tres nodes o punts de referència per a una localització en dues dimensions. Com que necessitem la utilització de tres nodes, aquests hauran d'estar sincronitzats de forma molt precisa i també el temps de referència.

#### 1.1.1.2. Temps diferencial d'arribada (TDoA)

És molt semblant a la tècnica de ToA però amb la diferència que es basa a determinar la posició de l'usuari o dispositiu mesurant les diferències de temps en diferents nodes receptors. Aquesta tècnica és molt comuna per als sistemes de navegació i es coneix com a navegació hiperbòlica. Quan mesurem la diferència en distància entre dos nodes receptors obtenim una sèrie de posicions infinites que satisfan aquesta mesura. Si tracem les possibles posicions, aquestes traces formen una corba hiperbòlica.

La sincronització entre els nodes receptors és un dels problemes d'aquesta tècnica. A més, un dels altres inconvenients és la necessitat que els nodes receptors han de ser capaços de mesurar amb precisió els temps d'arribada del senyal amb visió directa de l'emissor, ja que en interiors el senyal rebotarà i hi haurà efecte multicamí. Aquest efecte serà una de les dificultats per obtenir precisió, juntament amb el soroll additiu.

### 1.1.1.3. Angle d'arribada (AoA)

També anomenat direcció d'arribada, és una tècnica que es basa en la tècnica anterior, el temps diferencial d'arribada. Aquest mètode determina la TDoA que rep cadascun dels elements d'un *array* d'antenes receptores, i a partir d'aquests *delays* es pot calcular els angles d'arribada. Per a posicionament en 2D es necessiten dos senyals, i per a 3D es necessiten 3.

Un altre cop, l'inconvenient d'aquest mètode és l'efecte multicamí, per tant, és un mètode que difícilment es pot utilitzar en interiors. Els nodes receptors no necessiten estar sincronitzats, però cada node necessita *arrays* amb mesures de diferències de fase.

### 1.1.2. Mètodes basats en la potència de senyal

Els mètodes basats en la potència del senyal s'utilitzen tant en sistemes de comunicació de dades com sistemes basats en marcadors RFID. Aquest mètode es coneix com a RSS (*Received Signal Strength*), i es basa a mesurar la intensitat del senyal, ja que aquesta decau amb la distància. Molts dispositius mesuren la intensitat del senyal amb el sistema de mesura RSSI (*Received Signal Strength Indicator*). La mesura que obtenim va entre 0 i RSSI màxim, que pot acostuma a tenir uns valors de 100, 128 i 256.

La intensitat del senyal en el medi lliure decau teòricament amb el quadrat de la distància. L'equació que utilitzem per mesurar la distància d'un dispositiu amb aquest mètode és l'equació de transmissió de Friis, que és la següent:

$$\frac{P_r}{P_t} = G_t G_r \frac{\lambda^2}{(4\pi)^2 R^2} \quad (1.2.)$$

En la pràctica l'atenuació depèn dels efectes de l'atenuació deguda als efectes multicamí, reflexions, soroll. Aleshores els models realistes canvien  $R^2$  per  $R^n$  on  $n$  acostuma a estar entre 3 i 5.

### 1.1.3. Fingerprinting

És una tècnica de localització que es basa en el fet que les característiques del senyal de propagació són diferents en cadascun de les possibles localitzacions del dispositiu que es vol localitzar. Per tant, cada posició té una única empremta digital (*fingerprint*).

Es diu *fingerprinting* perquè el seu principi de funcionament és crear una matriu amb tots els senyals rebuts des de totes les posicions possibles dins de l'àrea de

funcionament del sistema. Les dimensions de l'empremta digital dependrà del nombre de sensors que es puguin escoltar.

Aquesta tècnica es divideix en dos passos:

- Fase calibratge: és la fase d'aprenentatge en què col·leccionem el conjunt de dades del senyal en forma de vector per a cada posició. La informació està formada per característiques del senyal, que dependrà de la informació que emetin els punts d'accés (ToA, TDoA, AoA, RSS). Cada *fingerprint* es recull en funció del temps. Aquesta fase s'haurà de repetir en cas que hi hagi un canvi important dins de l'àrea de cobertura del sistema, com pot ser una reorganització de l'espai.
- Fase real: el dispositiu o usuari a localitzar crea un vector amb tot el conjunt de dades del senyal que rep. Un cop tenim el nostre vector es compara amb tots els vectors d'informació que hem obtingut en la fase de calibratge mitjançant un algoritme que estima la localització òptima de l'usuari.

#### 1.1.4. Cell of Origin (CoO)

És un mètode de posicionament per a determinar la posició d'un dispositiu mòbil per la presència d'aquest dins d'una àrea. Amb aquest mètode només es pot saber dins de quina àrea o cel·la es troba el dispositiu. El funcionament es basa en la intensitat del senyal que emet el dispositiu a localitzar. La precisió d'aquest mètode depèn de la quantitat de nodes que formin cada cel·la i de la intensitat del senyal que emet el dispositiu. Normalment aquest mètode s'utilitza per aplicacions que no es necessita una precisió molt bona.

#### 1.1.5. Navegació per estima

La navegació per estima, en anglès, *dead reckoning* és un mètode de posicionament que es basa a calcular la posició tenint en compte la posició inicial i les velocitats acumulades amb el pas del temps. L'inconvenient d'aquest mètode és la inexactitud, ja que amb el pas del temps els errors es van acumulant i la precisió acaba sent dolenta.

#### 1.1.6. Filtre de Kalman

El filtre de Kalman és un algoritme que fusiona les posicions trobades amb la navegació per estima amb les posicions absolutes que provenen de qualsevol altre mètode. És un estimador de l'estat d'un sistema dinàmic lineal. És un mètode de posicionament molt potent que permet estimar variables d'estat que

no es poden observar a partir de variables observables que poden tenir qualsevol classe d'error.

El filtre de Kalman només es pot utilitzar per a sistemes lineals, per a sistemes no lineals s'utilitza el filtre de Kalman estès. Encara que linealitzar un sistema no lineal pot arribar a divergir, el filtre de Kalman estès és el mètode més utilitzat per a l'estimació d'estats no lineals en navegació.

## 1.2. Tecnologies posicionament

### 1.2.1. Wifi

El wifi és una tecnologia que permet la interconnexió sense fils entre dispositius electrònics (veure [11]). De fet, és una de les tecnologies sense cables més utilitzada. Els dispositius que tenen wifi es poden connectar entre si o a Internet. Hi ha diferents tipus de wifi però tots ells es basen en l'estàndard IEEE 802.11. Actualment hi ha dos tipus de wifi, el que treballa a una banda de 2,4 GHz i el que treballa a la banda de 5 GHz. El wifi que treballa a 5 GHz no té interferències, ja que no hi ha altres tecnologies que ocupin la mateixa banda.

A partir del wifi sorgeix el WPS (*Wifi Positioning System*) un sistema de posicionament basat en wifi. Per posicionar un dispositiu a través de WPS és necessari un conjunt de punts d'accés wifi (*hotspots*). Els mètodes més comuns per a posicionar un dispositiu amb l'ajuda d'aquests punts d'accés, són mesurar la intensitat del senyal dels *hotspots* rebut pel dispositiu i el *fingerprinting*. (veure [6], [12], [13], [14] i [15]).

El wifi permet l'enviament de dades d'un pes gran i a una velocitat molt alta, de fins a Gigabytes per segons.

El wifi com a sistema per a posicionament i navegació en interiors té una precisió dolenta per a espais molt petits, entre 2 i 15 m. A més, es necessita la instal·lació dels punts d'accés wifi per tota la zona on s'implementa el sistema, això fa que el sistema sigui costós quant a instal·lació, ja que tots els punts d'accés wifi requereixen subministrament elèctric. L'avantatge del wifi és el rang de cobertura que té, ja que és molt gran, major a 50 m i també la velocitat de transmissió de dades. A més, el wifi està molt utilitzat avui dia en el món i per tant hi ha un gran suport quant a aplicacions i millores de la tecnologia. A més els mètodes que s'utilitzen per a aquesta tecnologia són el *fingerprinting*, mètodes basats en la potència, mètodes basats en la triangulació (ToA, TDoA, AoA) i *Cell of Origin*.

### 1.2.2. Bluetooth Low Energy

El Bluetooth és una especificació industrial per xarxes sense fils d'àrea personal (WPAN) que possibilita la transmissió de veu i dades entre diferents dispositius mitjançant un enllaç radiofreqüència en la banda dels 2,4 GHz. El Bluetooth v4.0

és una especificació que inclou el Bluetooth clàssic, els d'alta velocitat i els de baix consum. Dins d'aquesta especificació de Bluetooth es troba el BLE o *Bluetooth Low Energy* (Bluetooth de baixa energia).

El BLE és un subconjunt de l'estàndard Bluetooth v4.0 que treballa amb un protocol nou per poder establir ràpidament connexions senzilles. Va ser dissenyat el 2001 amb l'objectiu d'aconseguir el mateix rang d'operació que el Bluetooth clàssic però utilitzant menys energia. El fet d'estalviar energia implica que la velocitat de transmissió de dades és una desena part i la latència és cinc vegades superior a la del Bluetooth clàssic. Per tant, el BLE no es pot utilitzar en aplicacions on s'hagin de transmetre una gran quantitat de dades. Malgrat això, la quantitat de dades que podem transmetre amb BLE és suficient per al posicionament en interiors.

Quant al BLE com a sistema de posicionament per interiors s'utilitzen uns petits *beacons* que són uns nodes petits i de baix cost. La instal·lació i el manteniment poden ser un esforç. Aquests *beacons* porten una bateria de botó, amb una autonomia molt gran. Pel posicionament amb BLE, els *beacons* utilitzen la tècnica de mesurar la distància a través de la intensitat del senyal que rep el dispositiu a localitzar. Els *beacons* transmeten senyal *broadcast*, és a dir, envia el senyal a tot el voltant. Té un rang d'entre 25 i 40 m, que dependrà de la potència del senyal que volem que enviïn els *beacons*. A més potència de senyal, més rang però menys duració de la bateria, i a menys potència de senyal, menys rang però més duració de la bateria. L'exactitud que es pot arribar a aconseguir amb BLE és d'aproximadament 1,5 m, però depèn del medi físic on instal·lem el sistema i la distància fins al dispositiu. Els dispositius com els telèfons tarden un temps a detectar que estan en el rang d'un *beacon*. El posicionament i navegació utilitzant el BLE és possible amb la implementació de mètodes basats en la triangulació, mètodes basats en la intensitat del senyal i *fingerprinting* (veure [16], [17], [18], [19]).

### 1.2.3. ZigBee

El ZigBee és l'especificació d'un conjunt de protocols d'alt nivell de comunicació basats en l'estàndard IEEE 802.15.4. Aquest estàndard fa referència a les xarxes sense fils d'àrea personal (WPAN). Està pensat per a la utilització amb ràdio fusió digital de baix consum. L'objectiu és l'enllaç segur entre dispositius que requereixen una taxa baixa d'enviament de dades i una vida útil molt gran de les bateries. El ZigBee està impulsat per una aliança anomenada ZigBee Alliance que està formada pels fabricants de dispositius electrònics més grans del món. Treballa a l'amplada de banda de 2,4 GHz.

Per a una xarxa amb aquesta tecnologia la xarxa pot ser d'estrella, arbre o malla, però la més utilitzada és l'organització en malla, d'aquesta forma els nodes de la xarxa estaran tots connectats a altres i en cas que falli un node la xarxa no estarà inoperativa. En una xarxa trobem tres categories de nodes.

- Coordinador ZigBee: és l'encarregat de controlar tota la xarxa, és el node més complet.
- Router ZigBee: interconnecta tots els nodes per executar el codi.
- Dispositiu: aquest és l'usuari, i només es connecta amb el node pare.

Els avantatges del ZigBee són la gran facilitat d'instal·lació i programació, ja que es necessita un 10% del hardware que es necessita en una xarxa Bluetooth i la meitat de codi. A més també té l'avantatge del baix cost i el baix consum energètic. Té una precisió d'uns 2 m, i el rang de cobertura en un espai indoor és d'uns 20 m. Els dos mètodes que s'acostumen a fer servir per aquesta tecnologia és RSSI i CoO (veure [20], [21], [22], [23], [24], [25]).

#### 1.2.4. Ultra Wide Band

L'*Ultra Wide Band*, abreviat com UWB, és una tecnologia de radiofreqüència que primerament s'ha utilitzat en el món militar, i ara s'utilitza en el món civil. És una tecnologia que es va començar a desenvolupar als anys 60 per empreses relacionades amb el món militar dels Estats Units.

Una transmissió en UWB consten d'una sèrie de micro impulsos d'energia modulats en picosegons. Aquesta durada dels impulsos tan petita fa que l'espectre en freqüència d'un senyal en UWB sigui molt ample, d'amplada de banda de varis GHz. Es considera que un senyal és UWB quan l'amplada de banda és major que 500 MHz o el 25% de la freqüència central. Com que l'UWB ha de distribuir l'energia sobre una amplada de banda tan gegant la densitat espectral d'aquesta energia és molt petita. Això ens permet enviar una gran quantitat de dades amb molt poca energia. UWB treballa en les freqüències que van dels 3,1 GHz fins als 10,6 GHz.

L'UWB està pensat per a treballar en freqüències on ja treballen altres tecnologies de connexió sense fils com el wifi o el Bluetooth sense causar interferències. Això és possible gràcies a la baixa potència de transmissió, repartida en una amplada de banda molt gran, fet que produeix una densitat espectral de potència molt baixa. Aquesta densitat espectral de potència es pot assimilar a la del soroll.

L'UWB té una velocitat de transmissió molt alta, normalment és de 480 Mbps però s'ha arribat a transmetre a 1,6 Gbps amb un rang de pocs metres. A més degut a la gran amplada de banda es pot transmetre una gran quantitat de dades.

Quant al posicionament utilitzant tecnologia UWB és una de les tecnologies que té millor precisió, al voltant dels 30 cm, fet que és un avantatge. Normalment utilitza la tècnica TDoA per a calcular la distància entre els nodes i el dispositiu a localitzar. Un altre avantatge de l'UWB és el baix consum energètic. Evidentment, l'UWB també té desavantatges, com el baix rang del senyal, que pot arribar a uns 10 m, si volem incrementar el rang, perdrem velocitat de transmissió. Un

altre desavantatge és la competència amb altres tecnologies de comunicació sense fils com el wifi i el Bluetooth, que estan molt implementades en el sector. Això fa que la implementació sigui més difícil. A més, el fet que no estigui molt implementat fa que el preu sigui molt elevat.

Per a poder utilitzar l'UWB com a tecnologia per a posicionament i navegació s'utilitzen els mètodes basats en triangulació, en la potència i el *fingerprinting*. A més, en la utilització de l'UWB el dispositiu a posicionar pot ser passiu o actiu. Si és passiu, l'element no té cap receptor ni emissor d'ones, es calcula la seva posició amb el mateix mètode que els radars, és a dir, uns receptors rebran el rebot de les ones en l'element a posicionar que han enviat els emissors d'UWB. En el cas que l'element a posicionar sigui actiu, serà l'element qui serà el receptor de les ones. (Veure [26], [27], [28], [29], [30], [31] i [32]).

### 1.2.5. RFID

RFID o identificació per freqüència (*Radio Frequency Identification*). El RFID és una tecnologia que permet identificar de forma remota, a través de xips, quin objecte està a sobre d'un senyal de ràdio. Per tant és una forma de comunicació sense fils entre lector i emissor. Es basa a intercanviar diferents freqüències de senyals de ràdio entre els lectors i els *tags*. El funcionament és molt senzill, un lector està enviant un senyal continu dins d'una àrea de cobertura, si un *tag* o etiqueta entra dins d'aquesta àrea de cobertura emet un senyal enviant una informació segons estigui programat. Aleshores el lector interpreta el senyal enviat per l'etiqueta i l'identifica. Per poder identificar-lo el lector està connectat a un servidor on té emmagatzemats tots els *tags* que pot reconèixer i el senyal que transmet cadascun d'ells. El lector i els *tags* utilitzen freqüències predefinides per poder enviar i rebre informació entre ells.

Un sistema RFID consta de dos tipus d'elements, els receptors i els emissors. Els receptors són els lectors, i els emissors són els *tags*. Els lectors tenen una antena, un transceptor, un processador i una interfície per poder connectar-se a un servidor. Els *tags* són uns microxips que tenen un circuit integrat i una antena. Els emissors poden ser actius o passius. Els *tags* passius no necessiten alimentació elèctrica mentre que els actius sí que necessiten.

La tecnologia RFID s'acostuma a utilitzar en processos on s'ha d'identificar una gran quantitat d'elements de forma periòdica. Tots els elements que s'han de poder identificar han de tenir un *tag*.

Quant a RFID per a posicionament es pot utilitzar, però té alguns inconvenients que ho dificulten. El principal problema per a posicionaments indoor amb RFID és el baix rang que tenen els emissors passius, ja que aquest rang és d'uns pocs centímetres, per tant necessitaríem molts lectors per poder cobrir un radi de cobertura d'una habitació. En canvi, si s'utilitza *tags* actius, el rang millora fins a la possibilitat dels 100 m però aleshores la precisió és molt dolenta, entre 1 i 3 m en el millor dels casos. En aquesta tecnologia els mètodes que podem utilitzar per a posicionament són pocs, tan sols es pot aplicar *fingerprinting*, *Cell of Origin*

i RSSI. Els mètodes basats en la triangulació són difícilment aplicables, i sempre s'haurien d'aplicar amb l'ajut d'algun altre mètode. (Veure [33], [34], [35], [36] i [37]).

### 1.2.6. Infrarojos

Les comunicacions sense fils amb infrarojos utilitzen espectre de llum invisible just per sota de la llum vermella visible. Les ones dels infrarojos són més llargues que les ones de la llum visible però més curtes que les ones de la radiació terahertz, més conegut com a raigs T. La llum infraroja és invisible per l'ull humà en la majoria dels casos, per tant, aquesta tecnologia és molt poc intrusiva pel posicionament indoor.

Per a posicionament els sistemes infrarojos acostumen a utilitzar un *beacon* que radia raigs infrarojos en totes les direccions, aleshores el dispositiu a localitzar utilitza dos receptors de raigs infraroigs que és capaç de mesurar l'angle. Els raigs infraroigs no poden sobrepassar les parets, per tant, per a cada habitació d'un espai indoor necessitarà un *beacon* emissor. L'exactitud d'un sistema de posicionament per infraroig és molt bona, pot arribar a tenir una precisió d'uns pocs centímetres, però el gran inconvenient és que té un rang d'uns pocs centímetres, no arriba a un rang d'un metre. Pel que fa al cost d'aquest sistema és molt reduït.

Els mètodes que es poden utilitzar són tan sols dos, angle d'arribada (AoA) i *Cell of Origin*. La precisió per aquests mètodes és molt variant per a CoO, l'exactitud és dolenta però el rang és gran. En canvi per a AoA l'exactitud és molt bona (d'uns pocs centímetres) però aleshores el rang de cobertura és molt petit. (Veure [38], [39], [40], [41] i [42]).

### 1.2.7. Ultrasons

Els ultrasons són ones acústiques que la seva freqüència està per sobre dels 20 kHz, és a dir, està per sobre de la capacitat de l'humà. Hi ha molts animals que utilitzen els ultrasons com a radar, les ones emeses pels animals reboten per l'espai i això fa que es puguin crear una imatge del seu voltant.

Basats en aquests animals es creen tres sistemes de posicionament indoor basat en ultrasons, aquests sistemes són *Active Bat*, *Cricket* i *Dolphin*.

- L'*Active Bat* és un sistema dissenyat per la universitat de Cambridge, està basat en el principi de la trilateració. Per a la localització es reparteixen pel sostre d'una habitació un conjunt de receptors d'ultrasons i l'objecte a localitzar porta un transmissor, anomenat Bat, que transmet ultrasons constantment. Aleshores a través de la tècnica ToA o ToF (temps de vol) es calcula la distància entre un node receptor i el Bat. Si



s'aconsegueix calcular les distàncies del Bat amb tres nodes receptors diferents, es pot determinar la posició del Bat en 3D. (Veure [43])

- El sistema *Cricket* funciona amb ultrasons i radiofreqüències. En aquest sistema l'estructura és la inversa a l'*Active Bat*. En aquest sistema el dispositiu a localitzar té un receptor, i els emissors estan distribuïts pel sostre d'una habitació. En aquest cas, la tècnica que s'utilitza per a calcular la posició és la TDoA. L'emissor envia una radiofreqüència i un pols d'ultrasons. Per tant, el receptor primer escolta una radiofreqüència i seguidament un pols d'ultrasons. (Veure [44]).

- El sistema *Dolphin* està basat en el mateix principi de posicionament que l'*Active Bat* i el *Cricket*. En aquest cas, també és un híbrid entre radiofreqüència i ultrasons. La radiofreqüència serveix per a sincronització entre els nodes i per enviar missatges entre aquests. L'estructura del sistema és igual que en l'*Active Bat*. Els nodes receptors estan distribuïts per l'habitació, i l'emissor té el dispositiu a localitzar. En aquest sistema els nodes tenen una CPU i tenen un comptador de polsos que serveix per mesurar la TDoA dels polsos d'ultrasons. Si tres o més nodes receptors són capaços de mesurar la distància amb l'emissor es pot posicionar en 3D el dispositiu. (Veure [45]).

Aquests tres sistemes tenen una precisió molt bona, de tan sols uns centímetres i també tenen un cost baix. El gran inconvenient d'aquests sistemes és que funcionen molt bé en habitacions petites. Quant a les dimensions són més grans que una habitació s'han d'introduir més nodes, fet que incrementar els costos notablement. A més en un espai més gran apareixen els efectes de soroll i *multipath*, això provoca que la precisió del sistema sigui dolenta. Els mètodes que es pot utilitzar per aplicacions de posicionament amb aquesta tecnologia són RSSI i alguns mètodes basats en la triangulació (ToA o ToF i TDoA) . (Veure [46], [47], [48] i [49]).

### 1.2.8. Basat en imatges

Els sistemes basats en imatges també anomenats òptics són totes aquelles tecnologies que utilitzen les imatges com a font principal per a poder posicionar qualsevol dispositiu o persona. Els sistemes de posicionament indoor basats en tecnologies òptiques es poden categoritzar en "ego-motion", és a dir, el sensor que s'utilitza està integrat i es mou amb l'element a localitzar i sensors estàtics, és a dir, el sensor està quiet i detecta l'element per posicionar (Veure [50]).

A part de les categories segons on es col·loca el sensor també es pot dividir segons el tipus de referència que pren el sensor pel posicionament.

- Referència sobre els models 3D: aquests sistemes prenen imatges i intenten identificar els objectes que surten a la imatge. Aleshores un cop s'ha identificat els objectes de la imatge es comparen amb la base de

dades on se sap on està cada objecte. Un cop es troba a la base de dades els objectes identificats es pot calcular la posició aproximada.

- Referència sobre imatges: aquests sistemes funcionen d'una manera molt semblant a l'anterior. El sensor pren una imatge i aleshores es compara aquesta imatge amb una base de dades d'imatges que s'han pres amb anterioritat. Aquesta base de dades conté imatges de tots els llocs on es vol instal·lar el sistema de posicionament. El sistema és capaç de trobar la posició gràcies a la comparació entre la imatge presa i les imatges de la base de dades.
- Referència d'objectius codificats implementats: el funcionament està basat en la identificació d'unes targetes amb un codi únic per a cada targeta. Amb la identificació de les targetes a través del sensor es calcula la posició.
- Referència d'objectius projectats: el funcionament és semblant a l'anterior, però en aquest en comptes de tenir les targetes codificades físiques, les targetes o objectius de referència estan projectats.
- Sense referència: el funcionament d'aquests sistemes es basa a veure com canvia de posició l'element a posicionar. És a dir, per a treballar sense referència és necessari treballar amb un sistema amb sensor estàtic.

Tots els sistemes òptics o basats en imatges per a posicionament en interiors poden arribar a tenir una precisió d'uns pocs centímetres. Tot depèn del sensor que s'utilitzi. Segons el tipus de referència que s'utilitzi l'esforç d'instal·lació i desenvolupament del sistema poden ser molt gran i el rang de cobertura del sistema pot variar molt. Quant al cost del sistema dependrà molt de la qualitat del sensor. Es pot desenvolupar sistemes òptics utilitzant una càmera d'un mòbil.

### **1.2.9. Basat en la inèrcia**

El sistema que es basa en la inèrcia com a principi fonamental pel funcionament per a la navegació, es diu Sistema de Navegació Inercial. Aquest sistema està format per giroscopis, acceleròmetres i un ordinador. Els acceleròmetres són sensors de moviment i els giroscopis són sensors de rotació. Per tant amb l'acceleròmetre es calcula acceleracions i amb el giroscopi es calcula la velocitat angular, que diu cap a quina direcció es mou. Amb aquest sistema som capaços de mesurar la posició, orientació i velocitat d'un objecte en moviment sense necessitat de tenir referències externes.

Aquest sistema s'utilitza per a navegar amb vehicles que no tenen contacte amb satèl·lits com submarins i míssils. El gran inconvenient del sistema inercial és el biaix que té aquest sistema. El biaix d'un estimador és la diferència entre l'esperança matemàtica i el valor del paràmetre que estima. Per tant, com més gran sigui el biaix inicial pitjor serà la posició determinada. A més, per trobar la

posició utilitzant aquest sistema s'integren els valors obtinguts del giroscopi i de l'acceleròmetre, per tant el biaix augmenta al quadrat i al cub en el llarg de trobar la solució a la posició. Això fa que durant el temps en el qual està funcionant el sistema el biaix vagi augmentant i per tant la posició determinada sigui cada cop pitjor.

Per tot això els sistemes inercials no s'utilitzen per usos domèstics, només per ús militar. Els sistemes inercials d'ús militar costen moltíssims diners però tenen un biaix pràcticament nul i per tant determinen la posició d'una manera molt precisa. Per aquest motiu no es pot utilitzar un sistema inercial com a única referència per al posicionament indoor. L'única possibilitat d'utilitzar el sistema de referència per a navegació indoor és la integració amb algun altre sistema, aleshores la posició determinada pot arribar a ser molt bona.

### 1.2.10. Lidar

Un Lidar (*Light Detection and Ranging*) és un dispositiu que permet mesurar distàncies des d'un emissor làser fins a un objecte utilitzant un feix de làser polsat. El Lidar està format per un focus emissor dels raigs làser infraroig i una lent receptora infraroja que és capaç de veure els feixos làser. Aquest feix de làser no és perillós per a la vista d'un humà. El funcionament es basa a enviar els raigs làser, aquests reboten en objectes i els raigs reflectits són captats per la lent receptora. D'aquesta forma el Lidar obté un mapa de punts que correspon a totes les reflexions que s'han produït en objectes i superfícies. Per aquest mapa de punts el més important és saber a la distància que es troben del Lidar, per mesurar la distància s'utilitza la tècnica ToA. Amb el mapa de punts creem un mapa en tres dimensions de l'entorn. El Lidar està pensat perquè funcioni donant voltes, d'aquesta manera som capaços d'analitzar l'entorn en els 360°. També hi ha alguns Lidars que són fixos. El Lidar s'està utilitzant molt en la navegació dels cotxes autònoms.

Els Lidars tenen unes prestacions molt bones, poden arribar a tenir unes precisions d'1 i 2 cm per a rangs de més de 100 m. A més de tenir unes molt bones prestacions, un dels avantatges és que mesuren en els 360° per tant amb un sol dispositiu creem una imatge en tres dimensions de tot l'entorn que ens envolta. El gran inconvenient del Lidar és el preu. Els Lidars que s'utilitzen per a l'automoció tenen preus de milers d'euros. Per a la navegació de drons i robots hi ha Lidar amb preus superiors als centenars d'euros. També es poden utilitzar Lidars fixos, que tenen una precisió d'uns pocs centímetres i un rang de desenes de metres. El preu dels Lidars fixos és d'uns 40 €. L'inconvenient d'aquests Lidars és que només mesuren la distància en una direcció i per poder trobar un núvol de punts de tot l'entorn necessitaríem més d'un dispositiu i aleshores el cost de configuració i desenvolupament és major (Veure [51], [52], [53], [54], [55] i [56]).

### 1.3. Elecció tecnologia

Un cop ja hem parlat sobre les principals tecnologies que s'utilitzen en el món del posicionament i navegació indoor recollirem en una taula tots els pros i contres de cadascuna de les tecnologies esmentades per a tal de tenir una visió global i per trobar més facilitats a l'hora de seleccionar la tecnologia en la qual basarem el treball. A més, també recullo tots els mètodes de posicionament aplicables per a cada tecnologia. En tots els mètodes serà aplicable també un filtre de Kalman, que servirà per a la fusió de dues o més tecnologies per dissenyar un sistema de navegació molt fiable.

**Taula 1.1.** Comparació entre tecnologies per a la navegació indoor

Tecnologia	Pros	Contres	Mètodes
<b>Wifi</b>	<ul style="list-style-type: none"> <li>- Rang de cobertura (&gt;50m)</li> <li>- Preu baix</li> <li>- Velocitat de transmissió</li> </ul>	<ul style="list-style-type: none"> <li>- Precisió (2-15m)</li> <li>- Instal·lació punts d'accés</li> </ul>	<ul style="list-style-type: none"> <li>- Fingerprinting</li> <li>- Triangulació</li> <li>- RSSI</li> <li>- CoO</li> </ul>
<b>BLE</b>	<ul style="list-style-type: none"> <li>- Rang de cobertura (&gt;25m)</li> <li>- Poc consum</li> </ul>	<ul style="list-style-type: none"> <li>- Precisió (&gt;1.5m)</li> <li>- Instal·lació nodes</li> </ul>	<ul style="list-style-type: none"> <li>- Fingerprinting</li> <li>- Triangulació</li> <li>- RSSI</li> </ul>
<b>ZigBee</b>	<ul style="list-style-type: none"> <li>- Poc consum</li> <li>- Rang cobertura (~20m)</li> <li>- Fàcil programació</li> </ul>	<ul style="list-style-type: none"> <li>- Precisió (&gt;2m)</li> </ul>	<ul style="list-style-type: none"> <li>- RSSI</li> <li>- CoO</li> </ul>
<b>UWB</b>	<ul style="list-style-type: none"> <li>- Precisió (~30cm)</li> <li>- Baix consum</li> </ul>	<ul style="list-style-type: none"> <li>- Cost</li> </ul>	<ul style="list-style-type: none"> <li>- Fingerprinting</li> <li>- Triangulació</li> <li>- RSSI</li> </ul>
<b>RFID</b>	<ul style="list-style-type: none"> <li>- Facilitat implementació</li> </ul>	<ul style="list-style-type: none"> <li>- Rang de cobertura</li> <li>- Precisió dolenta per a espais grans</li> </ul>	<ul style="list-style-type: none"> <li>- Fingerprinting</li> <li>- RSSI</li> <li>- CoO</li> </ul>
<b>Infrarojos</b>	<ul style="list-style-type: none"> <li>- Precisió (~cm)</li> <li>- Cost</li> </ul>	<ul style="list-style-type: none"> <li>- Rang (&lt; 1m)</li> </ul>	<ul style="list-style-type: none"> <li>- AoA</li> <li>- CoO</li> </ul>
<b>Ultrasons</b>	<ul style="list-style-type: none"> <li>- Precisió (~cm)</li> <li>- Cost</li> </ul>	<ul style="list-style-type: none"> <li>- Precisió bona per a rangs petits, precisió molt dolenta per espais més grans que una habitació.</li> </ul>	<ul style="list-style-type: none"> <li>- ToA o ToF</li> <li>- TDoA</li> <li>- RSSI</li> </ul>
<b>Òptica</b>	<ul style="list-style-type: none"> <li>- Precisió (~cm)</li> <li>- Cost</li> </ul>	<ul style="list-style-type: none"> <li>- Implementació difícil segons el mètode que s'utilitzi</li> </ul>	
<b>Inèrcia</b>	<ul style="list-style-type: none"> <li>- Fàcil instal·lació</li> </ul>	<ul style="list-style-type: none"> <li>- Precisió dolenta per a cost baix, precisió bona per a cost elevat</li> </ul>	
<b>Lidar</b>	<ul style="list-style-type: none"> <li>- Rang (~100 m)</li> <li>- Precisió (~cm)</li> </ul>	<ul style="list-style-type: none"> <li>- Cost</li> </ul>	<ul style="list-style-type: none"> <li>- ToA</li> </ul>

Un cop resumit en la taula 1.1 tots els avantatges i desavantatges de la utilització de les tecnologies esmentades amb anterioritat, la tecnologia que elegim per a desenvolupar un sistema de navegació indoor és l'òptica. Això és degut al fet que compleix un dels objectius principals del treball, que és que el sistema sigui de baix cost. A més d'aconseguir un cost reduït amb un sistema basat en imatges podem aconseguir una precisió d'uns pocs centímetres. L'elecció d'aquesta tecnologia també és degut a l'opció d'"ego-motion", ja que, aleshores la instal·lació de les càmeres només s'haurà de realitzar un cop i el sistema funcionarà per a tots els espais sense necessitat d'instal·lar res a cada espai.

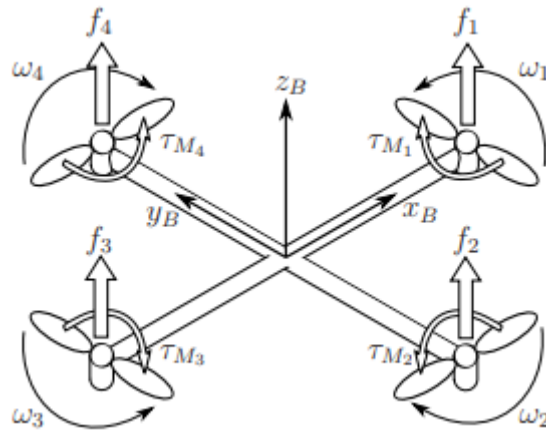
Per al desenvolupament d'aquest sistema de posicionament indoor utilitzarem una càmera de seguiment i una càmera de profunditat. Concretament utilitzarem dues càmeres Intel RealSense, que són càmeres desenvolupades per Intel. Aquestes càmeres estan pensades per a desenvolupadors i prototips de hardware. Tenen un rang d'aproximadament d'uns 10 m que pot augmentar depenent de les condicions de l'espai i la il·luminació. La precisió d'aquestes càmeres és de l'ordre de centímetres que depèn de la il·luminació de l'entorn que ens envolta.

## CAPÍTOL 2. MODEL MATEMÀTIC I CONTROL

Aquest capítol té l'objectiu de definir el model bàsic matemàtic d'un quadricòpter i les seves lleis de control. El desenvolupament d'un quadricòpter ha sigut molt recent perquè és molt difícil controlar quatre rotors independents sense l'ajuda d'assistència electrònica. Els quadricòpters tenen sis graus de llibertat i tan sols quatre controls d'entrada, per tant, és un sistema dinàmicament inestable i subestimat. Els sis graus de llibertat corresponen a moviments de translació i rotació. Aquests moviments es produeixen per la variació de la velocitat angular dels rotors del dron.

### 2.1. Sistemes de coordenades

Abans d'estudiar les lleis de control del quadricòpter és necessari establir un model matemàtic del dron. Per fer-ho assumirem que el quadricòpter és un sòlid rígid. En la figura 2.1 podem observar l'estructura d'un quadricòpter, on es veu la força que generen els quatre rotors, els moments i les velocitats angulars. Per compensar els moments que produeixen els motors, la rotació dels motors 1 i 3 són en sentit horari, i la rotació dels motors 2 i 4 són en sentit antihorari.



**Fig. 2.1** Estructura d'un quadricòpter

Abans de profunditzar en les físiques del moviment del dron s'ha de diferenciar entre les cinemàtiques dels sistemes de referència inercial i referència cos. La posició lineal absoluta del quadricòpter es defineix en el sistema de referència inercial amb les variables d'estat bàsiques  $x = [x \ y \ z]^T$  i la velocitat com  $\dot{x} = [\dot{x} \ \dot{y} \ \dot{z}]^T$ . L'actitud del dron respecte al sistema de referència cos queda definit pels angles *roll*, *pitch* i *yaw*,  $\theta = [\phi \ \theta \ \psi]^T$  i les velocitats angulars corresponents,  $\dot{\theta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ . S'ha de remarcar que el vector de la velocitat angular  $\omega \neq \dot{\theta}$ . Aquestes variables d'estat defineixen

l'actitud del dron respecte al seu propi sistema de referència. En el sistema de referència cos les velocitats linears estan determinades per  $V_B = [u \ v \ w]^T$  i les velocitats angulars per  $\omega = [p \ q \ r]^T$ . La velocitat angular és un vector apuntant a l'eix de rotació, mentre que  $\dot{\theta}$  és la derivada del temps del *roll*, *pitch* i *yaw*. Per convertir aquestes velocitats angulars a vector de velocitat angular, es pot utilitzar la següent fórmula, on  $W_\eta$  ve de [57] :

$$\omega = W_\eta \dot{\theta} \quad (2.1)$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.2)$$

Com veiem hi ha dos sistemes de referència. El sistema de referència inercial està centrat en la terra, es defineix la força de gravetat en l'eix z negatiu, aquest eix correspon amb l'eix de rotació de la terra. L'eix x sempre apunta a una estrella i l'eix y és perpendicular als altres dos. El sistema de referència cos està centrat en el dron i el seu centre de masses. Els eixos dels rotors del dron apunten a l'eix z positiu i els braços del dron apunten en els eixos x i y. La rotació d'un sistema de referència a un altre es fa a través de tres rotacions finites al voltant de tres eixos en un ordre específic. Utilitzarem la convenció Tait-Bryan, on primer girem respecte de l'eix z, després respecte de l'eix y i finalment sobre l'eix x. Les matrius de rotació respecte a cada eix són les següents:

$$R(z, \psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

$$R(y, \theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (2.4)$$

$$R(x, \phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \quad (2.5)$$

Per tant la matriu de rotació per passar de sistema de referència cos a sistema de referència inercial queda de la següent forma:

$$R = \begin{pmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \quad (2.6)$$

La matriu de rotació  $R$  és ortogonal, per tant,  $R^{-1} = R^T$ , que és la matriu de rotació de sistema de referència inercial a sistema de referència cos.

## 2.2. Equacions dinàmiques

Com hem dit abans assumim que el quadricòpter és un cos rígid i per tant, podem aplicar les equacions de Newton-Euler per modelar la dinàmica del dron. A més, també assumirem que el quadricòpter és simètric i el centre de masses coincideix amb el centre del dron, les hèlixs són fixes i l'impuls i el *drag* són proporcionals al quadrat de la velocitat dels rotors. També, per simplificar el problema no tindrem en compte els efectes aerodinàmics. Com podem veure en [58] i [59] les equacions de Newton-Euler per la dinàmica del dron són:

$$\begin{cases} F = m\dot{V}_B + (\omega \times mV_B) \\ \tau = I\dot{\omega} + (\omega \times I\omega) \end{cases} \quad (2.7)$$

La  $F$  correspon a la força total que actua en el centre de masses,  $\tau$  és el moment total que actua en el centre de masses,  $m$  la massa del dron,  $V_B$  la velocitat del cos,  $\omega$  la velocitat angular del cos i  $I$  el moment d'inèrcia en el centre de masses.

Les variables que ens interessin pel disseny d'un sistema de control són  $\dot{V}_B$ , canvis en la velocitat linear del cos, i  $\dot{\omega}$ , canvis en la velocitat angular del cos.

### 2.2.1. Forces

Començarem amb l'equació de la força. Per començar, les dues forces que actuen són l'impuls dels motors i la gravetat.

Per calcular l'impuls d'un sol motor es pot calcular de la següent forma, on  $C_T$  és el coeficient d'impuls,  $\rho$  és la densitat,  $r_p$  és el radi del rotor i  $A_r$  és l'àrea de la secció transversal de la rotació de l'hèlix:

$$T = C_T \rho r_p^2 A_r \omega^2 \quad (2.8)$$

Per simplificar l'equació utilitzarem una fórmula on l'impuls és proporcional al quadrat de la velocitat angular. No tindrem en compte les forces horitzontals en



les hèlixs ni la fricció induïda per l'aire en les direccions de l'eix x i y. A més, tampoc tindrem en compte els canvis en l'aerodinàmica que es produeixen en el moment de l'enlairament del dron coneguts com a efecte terra. La fórmula final de l'impuls per al sistema de referència cos és:

$$T = k\omega^2 = k \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 \omega^2 \end{bmatrix} \quad (2.9)$$

La força de la gravetat en el sistema de referència cos és de la següent forma:

$$G_B = R^T G_E = R^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = -mg \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (2.10)$$

Per tant, tan sols ens queda  $(\omega \times mV_B)$ , per aquesta fórmula s'utilitza la matriu de productes creuats per a la velocitat de rotació  $\Omega$ , que està definida de la següent forma:

$$\Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.11)$$

Aleshores l'efecte giroscòpic queda, tal com està en [60] :

$$\omega \times mV_B = m \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = m \begin{bmatrix} q\dot{z} - r\dot{y} \\ r\dot{x} - p\dot{z} \\ p\dot{y} - q\dot{x} \end{bmatrix} \quad (2.12)$$

D'aquesta forma ja tenim totes les variables per l'equació de forces de Newton-Euler:

$$k \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 \omega^2 \end{bmatrix} - mg \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} q\dot{z} - r\dot{y} \\ r\dot{x} - p\dot{z} \\ p\dot{y} - q\dot{x} \end{bmatrix} \quad (2.13)$$

Finalment, aïllant l'equació 2.13, aconseguim les equacions dinàmiques per les forces en el sistema de coordenades cos:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{k}{m} \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 \omega^2 \end{bmatrix} + g \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} - \begin{bmatrix} q\dot{z} - r\dot{y} \\ r\dot{x} - p\dot{z} \\ p\dot{y} - q\dot{x} \end{bmatrix} \quad (2.14)$$

### 2.2.2. Moments

Ara ens centrarem en l'equació de Newton-Euler que correspon als moments, l'objectiu és aconseguir les equacions dinàmiques pels moments expressades en el sistema de referència cos. Començarem amb totes les equacions dels moments que apliquen sobre el dron. Aquests moments són:

$$\tau = \tau_M + \tau_{\text{drag}} + \tau_{\text{gyro}} \quad (2.15)$$

Primer de tot tenim  $\tau_M$ , que fa referència als moments que indueixen els motors en l'eix x i y per als moviments de *roll* i *pitch*. La d es refereix a la distància entre el centre de masses del dron i els motors:

$$\tau_M = \begin{bmatrix} \tau_{\phi M} \\ \tau_{\theta M} \end{bmatrix} = \begin{bmatrix} d(T_2 - T_4) \\ d(T_1 - T_4) \end{bmatrix} = \begin{bmatrix} dk(\omega_2^2 - \omega_4^2) \\ dk(\omega_1^2 - \omega_3^2) \end{bmatrix} \quad (2.16)$$

Per calcular el moment induït pel moviment de *yaw* s'utilitza la següent fórmula, on  $C_Q$  és el coeficient de *drag*,  $\rho$  és la densitat,  $r_p$  és el radi del rotor i  $A_r$  és l'àrea de la secció transversal de la rotació de l'hèlix:

$$\tau = C_Q \rho r_p^3 A_r \omega^2 \quad (2.17)$$

Per simplificar l'equació, de la mateixa forma que hem fet abans amb l'equació de l'impuls, direm que el moment del drag és proporcional al quadrat de la velocitat angular, i b és el coeficient del moment. El moment total sobre l'eix z és causat per la suma de tots els moments de cada motor:

$$\tau_{drag} = b\omega^2 = b \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 \omega_i^2 \end{bmatrix} \quad (2.18)$$

Per últim, tan sols queda  $\tau_{gyro}$  que són els moments induïts pels efectes giroscòpics dels motors. Els moments giroscòpics tan sols afecten els eixos x i y. De l'equació 2.19,  $J_r$  és el moment d'inèrcia del dron, i  $\Omega$  és la suma de les velocitats angulars de cada motor:

$$\tau_{gyro} = \begin{bmatrix} \tau_{\phi,gyro} \\ \tau_{\theta,gyro} \end{bmatrix} = \begin{bmatrix} -J_r q \Omega \\ J_r p \Omega \end{bmatrix} = \begin{bmatrix} J_r q (\omega_1 + \omega_3 - \omega_2 - \omega_4) \\ J_r p (\omega_2 + \omega_4 - \omega_1 - \omega_3) \end{bmatrix} \quad (2.19)$$

Per tant la suma de moments del dron en el sistema de referència cos és:

$$\tau = \begin{bmatrix} dk(\omega_2^2 - \omega_4^2) \\ dk(\omega_1^2 - \omega_3^2) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} + \begin{bmatrix} J_r q (\omega_1 + \omega_3 - \omega_2 - \omega_4) \\ J_r p (\omega_2 + \omega_4 - \omega_1 - \omega_3) \\ 0 \end{bmatrix} \quad (2.20)$$

De l'equació (2.7) de Newton-Euler referida als moments tan sols ens queda  $I\dot{\omega} + (\omega \times I\omega)$ . De l'equació de Newton-Euler volem aïllar respecte  $\dot{\omega}$ . Per tant necessitem saber  $I$  i  $(\omega \times I\omega)$ .

$I$  és la matriu d'inèrcia. Una de les assumpcions que hem fet al principi és que el dron és simètric, per tant la matriu d'inèrcia serà una matriu diagonal:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.21)$$

$(\omega \times I\omega)$  és l'efecte giroscòpic dels moments:

$$\omega \times I\omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})qr \\ (I_{zz} - I_{xx})pr \\ (I_{xx} - I_{yy})pq \end{bmatrix} \quad (2.22)$$

Finalment l'equació dels moments de Newton-Euler queda:

$$\begin{aligned} & \begin{bmatrix} dk(\omega_2^2 - \omega_4^2) \\ dk(\omega_1^2 - \omega_3^2) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} + \begin{bmatrix} -J_r q \Omega \\ J_r p \Omega \\ 0 \end{bmatrix} = \\ & = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} (I_{yy} - I_{zz})qr \\ (I_{zz} - I_{xx})pr \\ (I_{xx} - I_{yy})pq \end{bmatrix} \end{aligned} \quad (2.23)$$

Finalment, aïllant l'equació 2.23, aconseguim les equacions dinàmiques pels moments en el sistema de coordenades cos:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{dk(\omega_2^2 - \omega_4^2)}{I_{xx}} \\ \frac{dk(\omega_1^2 - \omega_3^2)}{I_{yy}} \\ \frac{b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)}{I_{zz}} \end{bmatrix} + \begin{bmatrix} -J_r q \Omega \\ J_r p \Omega \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{(I_{yy} - I_{zz})qr}{I_{xx}} \\ \frac{(I_{zz} - I_{xx})pr}{I_{yy}} \\ \frac{(I_{xx} - I_{yy})pq}{I_{zz}} \end{bmatrix} \quad (2.24)$$

Finalment, de les equacions 2.14 i 2.24 podem extreure el model dinàmic del dron en el sistema de referència cos:

$$\left\{ \begin{array}{l} \dot{u} = -g \sin \theta - (q\dot{z} - r\dot{y}) \\ \dot{v} = g \cos \theta \sin \phi - (r\dot{x} - p\dot{y}) \\ \dot{w} = \frac{k}{m} \sum_{i=1}^4 \omega_i^2 + g \cos \theta \cos \phi - (p\dot{y} - q\dot{x}) \\ \dot{p} = \frac{1}{I_{xx}} \{ dk(\omega_2^2 - \omega_4^2) - J_r q \Omega - (I_{yy} - I_{zz})qr \} \\ \dot{q} = \frac{1}{I_{yy}} \{ dk(\omega_1^2 - \omega_3^2) + J_r p \Omega - (I_{zz} - I_{xx})pr \} \\ \dot{r} = \frac{1}{I_{zz}} \{ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) - (I_{xx} - I_{yy})pq \} \end{array} \right. \quad (2.25)$$

### 2.3. Sistema de control

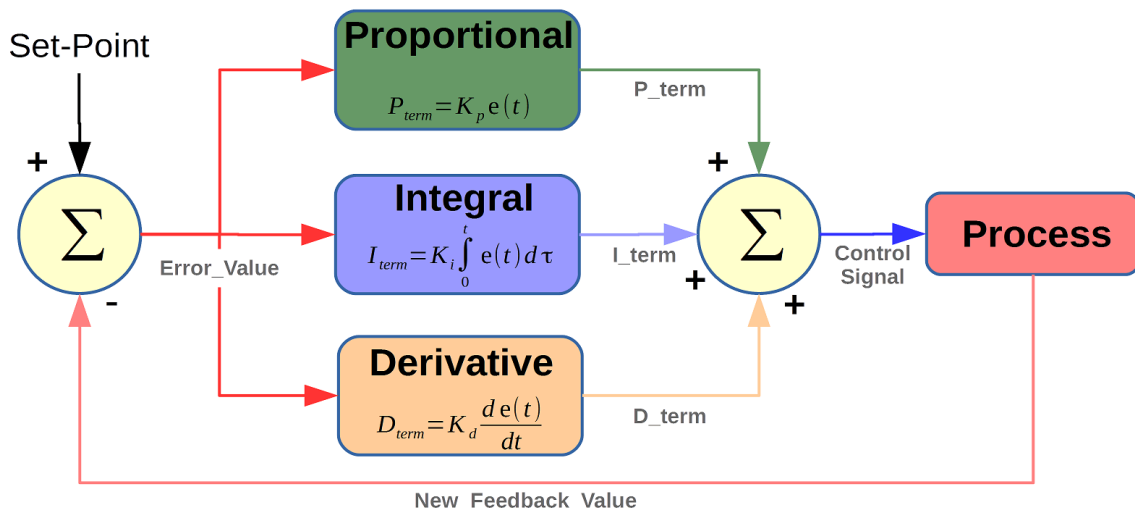
L'objectiu d'aquest apartat és entendre com funciona el sistema de control d'un dron. En aquest capítol tan sols veurem el funcionament d'un algoritme PID per un quadricòpter, ja que el disseny i simulació és una tasca amb un pes molt gran i no té cabuda dins d'aquest TFG, en el qual hi ha uns altres objectius definits.

El sistema de control està dissenyat a partir del model dinàmic del dron, és per això, que en l'apartat anterior hem estudiat aquest model. El sistema de control del dron té l'objectiu d'aconseguir un bon rendiment, i ser capaç de mantenir l'estabilitat quan estem sobrevolant un mateix punt o bé en seguir una trajectòria. En aquestes dues situacions el dron ha de ser capaç de fer el treball de forma correcta independentment de la pertorbació externa.

El sistema de control té informació d'entrada que els obté a partir dels sensors que té el dron o de les equacions dinàmiques i de les referències de les trajectòries. Les sortides del controlador són els senyals que s'envien als motors. Els senyals amb les que es controlen els motors són PWM (*Pulse Width Modulation*), aquest senyal transforma l'ordre del controlador amb velocitat angular.

El sistema de control d'un dron pot seguir diversos algoritmes de control com PID (*Proportional, Integral and Derivative controller*), LQR (*Linear Quadratic Regulator*), back-stepping control, *nonlinear  $H_\infty$  control*, etc. Els mètodes de control necessiten informació precisa per a poder funcionar de forma correcta. Estudiarem el funcionament d'un controlador PID a causa de la seva facilitat i robustesa.

El controlador PID és un mecanisme de control simultani per realimentació, calcula la desviació de l'error entre el valor mitjà i un valor que s'espera obtenir per aplicar una acció correctora. En la Figura 2.2 es veu l'estructura d'un controlador PID.  $K_p$  és el paràmetre proporcional,  $K_i$  el paràmetre integrador i  $K_d$  el paràmetre derivatiu. Mentre que  $e(t)$  és el senyal d'error.



**Fig. 2.2** Estructura d'un controlador PID

Els paràmetres estan correlacionats, és a dir, si es canvia el valor d'un paràmetre, els altres dos també s'hauran d'ajustar. El paràmetre proporcional és

el que s'encarrega de mantenir l'estabilitat, el paràmetre integrador s'encarrega de fixar el temps que tarda a repetir l'acció del proporcional i el paràmetre derivatiu és l'encarregat de suavitzar les reaccions. Els tres paràmetres han de ser els correctes per tal d'aconseguir un bon control del dron.

Com hem dit abans, un quadricòpter té sis graus de llibertat, tres graus de translació i tres de rotació, i tan sols té quatre variables de control, d'aquesta forma no és possible controlar el sistema, i per tant hem de fer una aproximació jeràrquica. El sistema de control es divideix en dos subsistemes de control, un bucle exterior i un bucle interior. Els moviments de rotació del dron es realitzen en tres angles, *pitch*, *roll* i *yaw*. Per cadascun d'aquests angles s'aplicarà un PID

El seu funcionament és molt simple, per tal d'explicar el funcionament ens basarem en la Figura 2.2. El procés és el que nosaltres volem aconseguir amb el controlador, aquest procés es mesura amb els components del dron, en aquest cas els giroscopis. Els valors obtinguts pels giroscopis han d'anar retroalimentant el nostre sistema, ja que d'aquesta forma el dron entén com es mou el dron en els seus eixos.

El "set-point" són els valors que nosaltres li manem fer al dron a través del ràdio control. Aquests valors entren al sistema de control PID. Aleshores el PID diferencia entre els valors que es reben dels giroscopis i els valors que vénen del ràdio control. A partir d'aquests dos valors es genera una diferència, anomenada "Error value", que nosaltres definim com  $e(t)$ . El valor de l'error és el que s'utilitza en cadascuna de les equacions del PID.

El Proporcional és el més important, ja que indica el nivell de correcció necessari a aplicar. El que fa és multiplicar  $e(t)$  pel paràmetre proporcional  $K_p$ . Aleshores el resultat de l'equació aplica un canvi que passa al senyal de control, que és el senyal PWM que va als motors. Aleshores en la següent interacció tornem a rebre els nous valors dels giroscopis, que es comparen amb les nostres ordres i el bucle torna a començar. En el cas que definim un paràmetre proporcional molt gran el senyal de control serà molt gran que podrà excedir el resultat que nosaltres volíem. Aleshores en la següent iteració haurem de corregir el senyal de control per tal de revertir l'excedent, i així succeirà de forma constant. Per tant, si el  $K_p$  és massa gran, produirà unes vibracions en el dron que no desitgem.

La part Integral s'encarrega de l'error acumulat, és a dir, l'error que tenim passat un cert temps. La deriva és l'error que s'utilitza en la fórmula de l'Integral. És per això, que es diu que aquesta part mira al passat. El resultat de l'equació Integral és un factor corrector de la deriva.

La part Derivativa s'encarrega dels errors ràpids. La part Derivativa entra en acció quan tenim moviments molt bruscos. Aquests moviments bruscos poden tenir l'origen en un valor de paràmetre proporcional massa elevat. Els moviments bruscos es traduiran amb una variació de  $e(t)$  molt gran entre diferents iteracions. El resultat de l'equació de la part Derivativa genera un factor de correcció per esmorteir els canvis ràpids (Veure [57], [58], [59] i [60]).

## CAPÍTOL 3. DISSENY DRON

Pel disseny d'un dron és necessari tenir clar quin és l'objectiu del dron i els usos que se li donaran. En aquest cas, el dron haurà de ser capaç de fusionar-se amb el sistema de navegació indoor que dissenyarem. Aquest sistema de navegació indoor estarà dissenyat per a ser instal·lat en qualsevol vehicle o dispositiu, en aquest cas el vehicle serà un dron.

Un cop ja sabem perquè volem el dron, l'elecció dels components que el componen hauran de ser triats amb molta precisió. Els drons es poden classificar segons el tipus d'ala i l'ús que li donem. Segons el tipus d'ala el dron pot ser d'ala fixa o ala rotatòria. Com que el dron que dissenyaré s'utilitzarà en espais interiors serà d'ala rotatòria, ja que aquests tenen enlairament vertical i són capaços de mantenir la posició. La sustentació es genera pel gir de les hèlixs. Els drons d'ala fixa aconseguixen sustentació de la mateixa forma que ho fan els avions. Quant a la classificació segons l'ús que li donem, acostumen a haver-hi de dos tipus, drons recreatius (drons per fotografia...) o drons de carreres. En aquest cas el dron serà recreatiu, ja que volem aconseguir un dron que sigui molt estable i no sigui difícil de controlar en espais petits (veure [61]).

Un cop ja sabem què volem aconseguir amb el dron i com ha de ser aquest, s'han de triar els components. Per fer-ho, ho farem de la mateixa forma que amb l'elecció de les tecnologies per al sistema de navegació, és a dir, cada component es compararà amb d'altres i es farà l'elecció segons les nostres necessitats, i evidentment també el preu.

### 3.1. Elecció dels components

#### 3.1.1. Estructura

L'estructura d'un dron, el fuselatge o el marc ha de ser molt fort perquè ha de suportar les forces dels motors, que a sobre van en sentits oposats. S'ha de mantenir sempre rígid i ha de ser resistent a alguns cops que pugui rebre el dron com en aterratges forçosos. El fuselatge també ha de ser capaç de mitigar les vibracions que produeixen el gir dels motors. Alhora el marc ha de ser lleuger per a facilitar l'elevació del dron, i ha de tenir espai per a poder portar el sistema de navegació que dissenyarem.

Amb la selecció de l'estructura del dron ja seleccionem el nombre de motors que tindrà el nostre dron. En el nostre cas l'estructura del dron serà per a quatre motors. Els drons que tenen quatre motors, es diuen quadricòpters. Els quadricòpters tenen moltes configuracions diferents, que es basen en la distància entre els eixos i també en la posició dels braços. Segons la posició dels braços les configuracions més típiques acostumen a ser en forma de X, en forma de + o en forma de H (veure [62], [63] i [64]). En el nostre cas utilitzarem la

configuració en forma de X, ja que, és la més utilitzada per la seva facilitat de configuració. A més, la distància entre eixos serà de 450 mm que és la distància que acostumen a utilitzar els drons per a fotografia, que són drons molt estables, i és el que volem aconseguir.

Els marcs acostumen a ser de plàstic o de fibra de carboni. Evidentment les estructures de fibra de carboni tenen molta més resistència i durabilitat, es mantenen rígids en tot moment. L'inconvenient dels marcs de fibra de carboni és que són més cars que les estructures de plàstic. En el cas d'una estructura en forma de X i amb distància entre eixos de 450 mm els preus dels fuselatges de fibra de carboni es poden trobar a partir de 60 €. En el cas de les estructures de plàstic tenen l'avantatge que acostumen a fabricar-se amb una impressora 3D i per tant, és molt fàcil trobar recanvis i són molt assequibles. Els preus dels marcs de plàstic amb configuració en forma de X i una distància entre eixos de 450 mm es poden arribar a trobar a partir dels 20 €. En el nostre cas elegirem un marc de plàstic, ja que té un preu més baix i segueix cobrint les necessitats pel meu dron.

### 3.1.2. Controlador de vol

El controlador de vol és el cervell del dron. És una placa amb circuits integrats que té sensors que detecten canvis durant el vol, aquests sensors acostumen a ser giroscopis, acceleròmetres, baròmetre i magnetòmetre. A més els controladors de vol són el punt de connexió de tots els elements del dron. A part de rebre la informació que obtenen dels sensors també reben les ordres de l'usuari.

Com que el nostre objectiu és utilitzar el dron per integrar-li el sistema de navegació indoor que dissenyarem, el dron ha de ser estable per tant el controlador de vol que necessitem és un que estabilitzi el dron i a més, que siguem capaços de fer-li modificacions de software, ja que serà necessari per a la utilització del sistema de navegació. A part de tot això, el controlador de vol haurà de ser econòmic.

Principalment els controladors de vol es divideixen segons l'ús que li donem al dron. Hi ha controladors de vol per a carreres, per a fotografia i per a vol autònom. El controlador de vol que necessitem serà un dels que s'utilitzen per aplicacions de fotografia o cinema i que sigui capaç de fer vol autònom, per tant els controladors de vol per a drons de carreres queden descartats (veure [65], [66], [67] i [68]).

Els controladors de vol que s'acostumen a utilitzar per aplicacions d'aquest tipus són de la marca DJI, Pixhawk o APM. Els controladors de vol de la marca DJI tenen una estabilitat molt bona, són molt fàcils d'instal·lar en el dron i no necessiten una configuració prèvia molt gran. Els controladors de vol DJI són els que s'utilitzen en aplicacions de cinema i fotografia professional. Un dels inconvenients d'aquests controladors de vol són el seu preu, ja que costen a partir de 100 €. A més, el gran inconvenient d'aquests controladors és que no es pot fer cap modificació en el codi. Per aquest últim motiu descartem la utilització



d'aquests controladors. Per tant l'elecció queda entre un controlador de vol Pixhawk i un APM. L'avantatge d'un Pixhawk enfront d'un controlador APM és que el Pixhawk està actualitzat mentre que APM ja està obsolet. A més, té més opcions i més capacitat i una electrònica millor. A part, la comunitat de suport de Pixhawk és molt més gran i està més activa.

El controlador de vol Pixhawk que utilitzarem serà Pixhawk 2.4.8 que té un processador de 32 bits, amb una RAM de 168 MHz. Té giroscopis, acceleròmetres, magnetòmetre i baròmetre com a sensors incorporats. Es pot trobar a partir d'uns 40 €.

### 3.1.3. Motors

Els motors són una de les parts més important del dron. Són els encarregats d'aixecar, mantenir i volar en la direcció que nosaltres volem. És molt important que escollim els motors adequats perquè d'ells dependrà l'estabilitat del nostre dron. Els motors hauran de ser capaços de lliurar la mateixa potència perquè si no aleshores tindrem problemes d'estabilitat. Els motors es poden classificar segons la seva velocitat màxima de rotació i el tipus de motor.

Els motors que s'acostumen a utilitzar són "*brushed*" i "*brushless*". Els motors *brushed* són els motors que tenen escombretes i els *brushless* no en tenen. Els motors *brushed* tenen el bobinat a la part mòbil, per això necessita les escombretes per a passar l'electricitat. Els motors *brushless* tenen el bobinat estàtic i la part mòbil és un conjunt d'imants. Els motors *brushed* tenen l'inconvenient que les escombretes van fregant amb la part mòbil i això produeix un desgast. Amb aquest desgast es perd eficiència. En canvi els motors *brushless* com que no tenen cap fregament no tenen aquest desgast i per tant no perden eficiència. Com que no hi ha fregament el rendiment d'aquests motors és millor que en el cas dels motors *brushed*, per tant s'aconsegueixen potències més grans amb menys consum. El preu dels motors *brushless* és una mica més elevat, ja que són més complicats de fer-los petits, per a drons, però els avantatges són significatius (veure [69], [70] i [71]). És per això, que pel nostre dron escollirem motors *brushless*.

L'altre paràmetre a analitzar a l'hora d'elegir un motor pel dron és el número de kV del motor. kV significa 1000 revolucions per V que rep el motor. Els motors s'escullen segons el tipus de vol que farem amb el nostre dron. Per a drons de fotografia s'utilitzen motors amb menys de 1200 kV, per a drons de carreres o acrobàtics s'utilitzen drons a partir de 1200 kV. Un motor amb kV baix voldrà dir que el seu moviment és més suau, no són motors molt vius i el consum elèctric és menor. En canvi per a motor amb kV alt les seves reaccions seran més agressives, ja que es tracta d'un motor més viu. A més, els motors amb kV alt tenen un consum d'energia major (veure [70], [71] i [72]).

Com que volem que el dron sigui molt estable, amb moviments suaus, elegirem un motor amb kV baix. Encara que tingui un kV baix els motors han de tenir suficient empenta per a poder aixecar el dron i mantenir-lo volant. Es considera

que els motors dels drons han d'aguantar cadascun la meitat del pes del dron, per tant, quatre motors tindran l'empenta igual al doble del pes del dron (veure [73]).

Un cop ja tenim analitzats totes les possibilitats de motors de drons l'elecció serà més fàcil. Escullim un motor *brushless* de 920 kV. Segons les especificacions d'aquest motor és capaç d'aixecar un pes d'1,4 kg. Per tant, serà suficient per aixecar el dron amb el sistema de navegació incorporat. Encara que l'elecció dels motors ha de ser l'adequada l'elecció de la bateria i el controlador de velocitat seran claus perquè els motors treguin el màxim rendiment.

### 3.1.4. Controlador de velocitat

El controlador de velocitat o ESC (Electronic Speed Control) és un controlador de velocitat electrònic. És el sistema que s'encarrega de definir la velocitat de gir dels motors *brushless* del dron. Defineix la velocitat de gir del motor a través de la generació d'uns polsos compatibles amb els motors del dron. Els ESCs és l'element que connecta els motors amb el controlador de vol. Totes les ordres que li donem al controlador de vol, les transmet als motors.

Els ESCs es classifiquen segons la quantitat de corrent que poden subministrar de forma constant al motor. Per poder seleccionar els ESCs de forma correcta ens hem de fixar en les especificacions dels motors que hem escollit. D'aquesta manera podrem esbrinar el consum de corrent màxim dels motors. A més, en les especificacions dels motors ens dirà quin tipus de bateria necessitem. Per poder volar el dron sense problemes els quatre ESCs han de ser iguals, perquè si no el vol serà més erràtic i el dron serà més difícil de maniobrar (veure [64], [74], [75] i [76]).

Un cop ja sabem els paràmetres que hem de tenir en compte per a l'elecció dels controladors de velocitat podem fer la tria adequada. Com que els motors que hem escollit són de 920 kV, segons les seves especificacions els ESC que funcionen bé són els que tenen suporten almenys 11,1 V.

### 3.1.5. Bateria

Les bateries dels drons poden ser de molts tipus, actualment s'utilitzen les bateries Li-Po, que són bateries de polímer de liti. Són les bateries més modernes, el component amb què les fabriquen permet aprofitar els espais al màxim. Aquestes bateries es fabriquen per cel·les i cada cel·la té un voltatge de 3,7 V (veure [64], [74], [75], [76] i [77]).

Per l'elecció de la bateria ens tornarem a fixar en les especificacions dels motors escollits. En les especificacions ens marquen la utilització d'una bateria de tres cel·les, per tant, el voltatge de la bateria serà d'11,1 V.

En elegir a la bateria per tant, ens haurem de fixar en un parell de coses més, la capacitat i el pes. A més capacitat més pesarà i aleshores el dron consumirà l'energia més ràpida. A menys capacitat, menys pesarà però aleshores l'energia es consumirà molt de pressa. Per tant s'ha d'agafar una bateria que es trobi en un punt entremig. Per escollir la bateria utilitzarem la regla de l'empenta dels motors. Els motors han de suportar el doble del pes del dron, per tant, el pes de la bateria no ha de sobrepassar aquest límit. La bateria que hem escollit té una capacitat de 4000 mAh i un pes de 300 grams aproximats.

### 3.2. Resum dron

Un cop ja tenim tots els elements vitals del dron seleccionats es recullen en una taula el model de l'element seleccionat i el seu preu, per poder tenir una visió global del dron que utilitzaré.

A més dels elements definits en els apartats anteriors el dron està format per uns altres elements com pot ser la telemetria, el mòdul GPS, el receptor i transmissor de ràdio, el *buzzer* i el *switch*.

**Taula 3.1.** Resum elements dron

Element	Model - Especificacions	Preu
<b>Estructura</b>	F450	20 €
<b>Controlador de vol</b>	Pixhawk 2.4.8 + hèlixs + Mòdul GPS + cables	80 €
<b>Motor</b>	2212 920 kV	20 €
<b>ESC</b>	Simonk 30A	18 €
<b>Bateria</b>	Bateria Li-Po U-TECH 3s 4000 mAh 11,1 V	30 €
<b>Telemetria</b>	3DR Telemetria 433 MHz	22 €
<b>Receptor i emissor ràdio</b>	Flysky FS-i6S Emisor 10 CH Receptor 10 CH	55 €

El resultat de muntar totes les peces que s'han comprat es pot veure en Figura 3.1, on es veu el resultat final del dron construït.



**Fig. 3.1** Imatge del dron amb les peces seleccionades

Per saber tots els passos que s'han seguit per al muntatge del dron es poden veure en l'apartat "Muntatge dron" dins de l'annex. De la mateixa forma, per veure la configuració que s'ha aplicat al dron es pot veure l'apartat de l'annex nomenat "Configuració dron".

### 3.3. Primer vol

Ara que està el dron amb tot el hardware instal·lat i configurat arriba la comprovació de tot el treball fet fins ara. En el nostre cas es realitzaran dues proves de vol. Una prova serà en exteriors amb senyal GPS i l'altra serà en un espai interior on no hi ha senyal GPS.

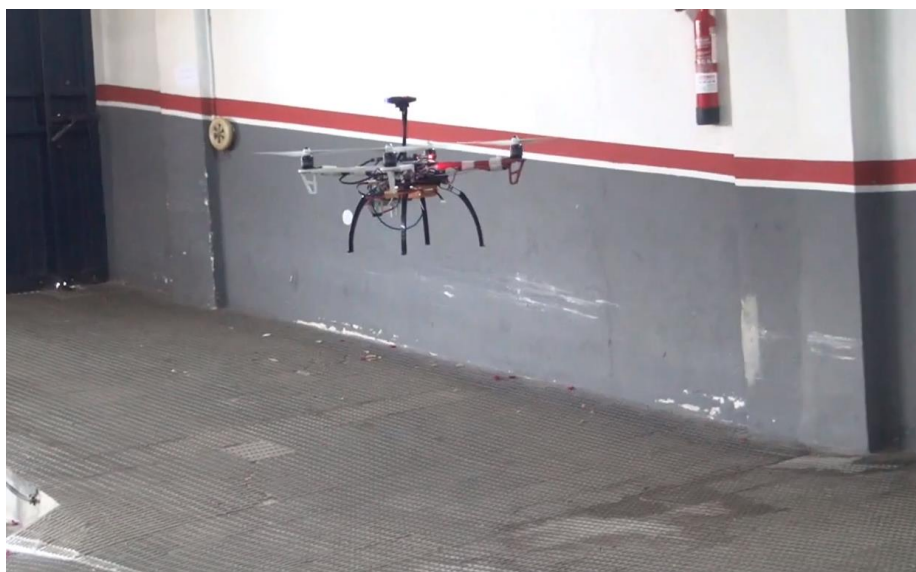
El vol en un espai exterior amb senyal GPS s'ha realitzat sense cap problemes, el mòdul GPS ha sigut capaç d'obtenir senyal GPS i per tant s'ha posicionat de forma correcta. Quan el dron ha volat en el mode estabilitzat aquest ha mantingut la seva posició. Durant la realització d'aquesta prova de vol hi ha hagut vent. Gràcies al posicionament per GPS, el dron ha lluitat en contra el vent per a mantenir-se quiet a l'aire. Per poder veure els resultats del primer vol en exterior s'ha adjuntat un enllaç del vídeo gravat, es pot veure al **punt 3.1. dels annexos**.

En la figura 3.2. es pot veure el dron volant durant la prova de vol.



**Fig. 3.2** Imatge del dron durant la realització del vol amb senyal GPS en un espai exterior

La segona prova ha sigut el vol en un espai interior on no arriba el senyal GPS. Els resultats de la segona prova de vol són molt diferents. Un cop el dron s'ha elevat amb el mode estabilitzat el dron no ha sigut capaç de mantenir la posició igual que en la primera prova de vol. Durant la prova de vol s'ha mogut el dron a través del ràdio control, quan deixàvem d'enviar-li ordres de moviment al dron aquest es seguia movent de forma més lenta. Aquest moviment és causat per la manca del senyal GPS, ja que sense cap sistema addicional no som capaços de mantenir la posició. La prova de vol es pot veure en el vídeo adjuntat en el **punt 3.2. dels annexos**.



**Fig. 3.3** Imatge del dron durant el vol en un espai interior sense senyal GPS

Com a conclusió dels primers vols amb el dron veiem que en un espai sense senyal GPS és impossible que el dron mantingui la posició de forma autònoma. En canvi, en un espai exterior amb senyal GPS el dron lluita contra l'aire per mantenir la posició. Com a resultat de la segona prova de vol veiem la necessitat de dissenyar un sistema de navegació per a interiors.

## CAPÍTOL 4. DISSENY SISTEMA DE NAVEGACIÓ

Un sistema de navegació, és un sistema que ens permet posicionar dins d'un espai i a més ens permet saber com anar des d'un punt A fins a un punt B. En el nostre cas el sistema de navegació serà per espais que es troben sense senyal GPS, que és el sistema de navegació per satèl·lit més utilitzat en el món. Com ja hem explicat en el primer capítol de la memòria el sistema de navegació utilitzarà la tecnologia òptica com a font per al posicionament i la navegació. Per tant aquest capítol està enfocat en l'explicació del desenvolupament del sistema de navegació basat en dispositius òptics.

L'odometria és l'estimació de la posició relativa d'un vehicle en l'espai. És una estimació, ja que, és impossible determinar la posició de forma exacta. I és la posició relativa, ja que, la posició serà relativa al punt inicial de navegació. Com que la font d'informació que utilitzarem nosaltres seran càmeres, aquesta tècnica es coneix com a odometria visual.

Aquest capítol està dividit en quatre apartats, el primer de tot serà un capítol on s'exposarà les eines hardware que utilitzarem. El segon apartat tractarà de les eines software que s'utilitzen i en el tercer apartat explicarem de forma detallada el procediment del sistema que dissenyem. Per últim es mostraran els resultats obtinguts de volar el dron amb el sistema de navegació.

### 4.1. Eines hardware

#### 4.1.1. Intel RealSense

Com a eines hardware per al sistema de navegació utilitzarem dues càmeres d'Intel, tal com havíem explicat en el capítol 1. Intel RealSense és la tecnologia dissenyada per l'empresa americana que està dissenyant dispositius amb capacitat per veure l'entorn que l'envolta, entendre'l, aprendre d'ell i saber interactuar-hi. La tecnologia Intel RealSense ofereix càmeres amb sensor de profunditat i també càmeres de seguiment. Pel disseny del nostre sistema utilitzarem una càmera amb sensor de profunditat i una altra de seguiment. Un dels principals avantatges d'aquesta tecnologia és la gran facilitat de començar a utilitzar i aprendre d'ella.

##### 4.1.1.1. Càmera amb sensor de profunditat

La càmera amb sensor de profunditat escollida és la Intel RealSense D435. Aquesta càmera és una de les més avançada dins de la família de la sèrie D400. El cervell de la càmera és el processador de visió Intel RealSense D4, que és l'encarregat de processar les imatges que pren la càmera i processar mapes en

3D amb una gran facilitat i sense necessitar un processador extern o GPU dedicada.

La càmera té quatre elements per a l'obtenció de les imatges. Té dos sensors infrarojos que capturen la imatge, un a l'esquerra i un altre a la dreta. També té un projector IR (Infrarojos) per tal de seguir veient en condicions amb poca visibilitat. I per últim té un mòdul RGB per tal d'obtenir el color de les imatges preses (veure [78]).

A partir dels dos sensors IR que capturen imatges es pot extreure la profunditat d'una imatge. La precisió és bona, ja que, es comparen les imatges que obtenen cadascun dels sensors. Quant a les característiques tècniques de la càmera, els sensors IR que mesuren la profunditat treballen a 90 FPS, mentre que el mòdul RGB treballa a 30 FPS. La D435 treballa en un rang d'entre 0,11 m i 10 m segons el *datasheet* però aquests valors varien segons les condicions de treball on es troba la càmera. La resolució és de fins a 1280x720.

L'avantatge d'aquesta càmera respecte a les altres de la seva família és el mòdul dels components, ja que els components d'aquesta càmera són més precisos.

Aquesta càmera l'utilitzarem per a detectar objectes que es trobin en la nostra ruta. A partir de la localització dels objectes podrem saber la distància a la qual ens trobem respecte ell per tal de protegir-nos.

#### **4.1.1.2. Càmera de seguiment**

La càmera de seguiment és la Intel RealSense T265, fins al moment és l'única càmera de seguiment que disposa Intel dins de la tecnologia RealSense. El cervell de la càmera és la unitat de processament de visió Intel Movidius Myriad 2.

La càmera té dos sensors d'ull de peix i un IMU (Inertial Measurement Unit). Aquests sensors i l'IMU són les úniques fonts d'informació que té la unitat de processament per tal d'obtenir una solució de seguiment (veure [79]).

La tècnica que utilitza aquesta càmera és V-SLAM (*Visual Simultaneous Localization And Mapping*). SLAM és un algoritme que és capaç de realitzar un mapejat de l'espai on estem i alhora és capaç de navegar per ell. Un algoritme SLAM està format per cinc passos: inicialització, localització, mapejat, relocalització i finalment optimització del mapejat. El V-SLAM és un algoritme en què les fonts d'informació de l'algoritme provenen de fonts visuals, en aquest cas la càmera T265 (veure [80]).

El funcionament de la càmera és molt senzill, a través dels sensors òptics i la unitat inercial que té la càmera s'obté la informació, que s'utilitza en l'algoritme V-SLAM que s'executa directament en la unitat de processament de la càmera. Els resultats que dona la càmera és la informació per als sis graus de llibertat que tenim. Els sis graus de llibertat són tres graus de translació i tres graus de



rotació. Els tres graus de translació són: endavant/enrere, esquerra/dreta i dalt/baix. Els tres graus de rotació són: *pitch*, *roll* i *yaw*.

El punt fort de la càmera T265 és que com el desenvolupament del V-SLAM és propi d'Intel funciona de forma molt eficient en la unitat de processament de la càmera i alhora amb una precisió molt bona.

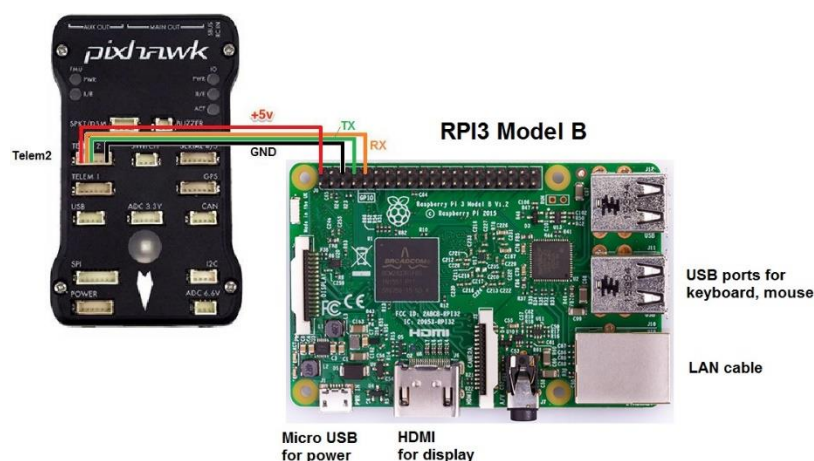
#### 4.1.2. Ordinador de bord

Per tal gestionar tota la informació que es rep de les càmeres Intel RealSense s'utilitzarà una Raspberry Pi 3B. La Raspberry Pi és un ordinador de placa reduïda de baix cost. La Raspberry Pi 3B té 1 GB de RAM i una CPU de 64bits a 1.2 GHz de quatre nuclis, l'ARMv8. Permet la connexió Wifi (802.11n), Bluetooth 4.1 i Ethernet fins als 300 Mb/s.

La Raspberry Pi és l'encarregada de tenir instal·lat l'aplicació que hem dissenyat per al sistema de navegació indoor i també, és el punt d'obtenció de dades de les càmeres Intel RealSense.

Les dades que s'obtidran de l'execució del programa dissenyat per al sistema de navegació indoor, es poden utilitzar en qualsevol classe de vehicle, ja que no l'hem dissenyat per a un de concret. En el nostre cas el sistema de navegació indoor s'utilitzarà en un dron, per tant, les dades que obtenim de la Raspberry Pi s'hauran de traspasar al controlador de vol del dron.

Per fer la connexió entre la Raspberry Pi i el controlador del vol es pot fer a través dels ports UART de la Raspberry, tal com es pot veure en la Figura 4.1, i també es pot fer a través d'USB.



**Fig. 4.1** Connexió entre la Raspberry Pi i la Pixhawk 2.4.8 a través dels ports UART

## 4.2. Eines software

### 4.2.1. Llenguatge de programació

El llenguatge de programació que utilitzarem serà el Python. El Python va aparèixer el 1991, l'objectiu principal era convertir-se en un llenguatge de programació que fos fàcil d'utilitzar i amb una filosofia molt clara, que fos un llenguatge amb una sintaxi que afavoreixi el codi llegible. A més, el Python és un llenguatge que té una corba d'aprenentatge molt suau. Tot això junt amb el fet que sigui gratuït inclús per a fins comercials, fa que Python s'hagi convertit en uns dels llenguatges de programació més utilitzats del món. De fet, grans empreses com Google, Facebook, Spotify o Netflix l'estan utilitzant.

Un dels grans avantatges de Python és que és considerat un llenguatge de programació general, ja que es poden desenvolupar pàgines web, aplicacions, i programes d'intel·ligència artificial i *Data Science*. A més, Python té una gran quantitat de llibreries que es poden descarregar de forma gratuïta i molt fàcil (veure [81] i [82]).

L'IDE (*Integrated Development Environment*), és a dir, l'entorn de desenvolupament que hem utilitzat per crear el projecte ha sigut el PyCharm, de l'empresa txeca JetBrains. Aquest IDE està especialitzat en el llenguatge de programació Python.

L'elecció del Python com a llenguatge de programació es deu a la gran facilitat d'aprenentatge que té. També un dels motius per triar Python és la gran comunitat que l'utilitza i per tant el suport que hi té. Per últim i el motiu amb més pes que fa que ens haguem decantat pel Python és que la llibreria que proporciona Intel per treballar amb les càmeres RealSense és compatible amb Python.

### 4.2.2. Comunicació amb les càmeres

Per la comunicació amb les càmeres RealSense, Intel ha creat un kit de desenvolupament de software (SDK) perquè els desenvolupadors puguin crear un projecte informàtic treballant amb aquestes càmeres. El SDK (Software Development Kit) es diu Intel® RealSense™ SDK 2.0, i és una llibreria multiplataforma que dóna suport tant a les càmeres de profunditat de la família D400 i SR300 com a la càmera de seguiment T265.

Aquests SDK emmagatzema les llibreries pels diferents llenguatges de programació que suporten les càmeres i a més a més, inclou tota la documentació per a poder treballar amb les càmeres RealSense. També incorpora un conjunt d'exemples per a cada llenguatge de programació i per a cada tipus de càmera (veure [83]).

La llibreria de Python es diu `librealsense2` i es pot instal·lar a través de l'IDE que utilitzem o a través de l'instal·lador que ens proporciona el SDK. El gran avantatge de la llibreria és que és codi obert i per tant els desenvolupadors podem proposar canvis en el codi. Això fa que gràcies a l'aportació d'altres usuaris hi hagi un gran nombre d'exemples extra que fa que puguem veure moltes configuracions per a la utilització de les càmeres, i també dubtes sobre el seu funcionament.

Per poder treballar amb les càmeres a la Raspberry Pi que serveix com a ordinador de bord del dron haurem d'instal·lar correctament la llibreria seguint les instruccions per al sistema operatiu de la Raspberry Pi.

### 4.2.3. Comunicació amb el pilot automàtic

En aquest apartat parlarem de les dues eines que utilitzem per a la comunicació amb el pilot automàtic. Una eina s'encarrega de la comunicació entre el controlador de vol del dron (Pixhawk 2.4.8) i el GCS (*Ground Control Station*). Aquesta eina és el MAVLink, que bàsicament és un protocol de comunicació. L'altra eina que utilitzem és el Dronekit, que és la llibreria escrita en Python que s'encarrega de la comunicació entre l'ordinador de bord (Raspberry Pi) i el controlador de vol del dron.

#### 4.2.3.1. Protocol MAVLink

MAVLink és l'acrònim de *Micro Aerial Vehicle Link*, és un protocol de comunicació que està específicament dissenyat per a la comunicació amb micro UAVs (veure [84] i [85]). A partir del protocol sorgeix una llibreria que treballa seguint el protocol. La llibreria és molt lleugera i conté la informació necessària per a l'intercanvi de missatges. Aquesta informació necessària està continguda dins d'arxius `.xml` d'aquesta manera la llibreria es pot utilitzar amb diferents llenguatges de programació.

La comunicació a través del protocol MAVLink funciona a través de l'enviament de paquets. Actualment hi ha dues versions de MAVLink la v1.0 i la v2.0 però la més utilitzada de moment és la primera versió. L'estructura d'un paquet del protocol de la primera versió es pot veure en la figura 4.2.

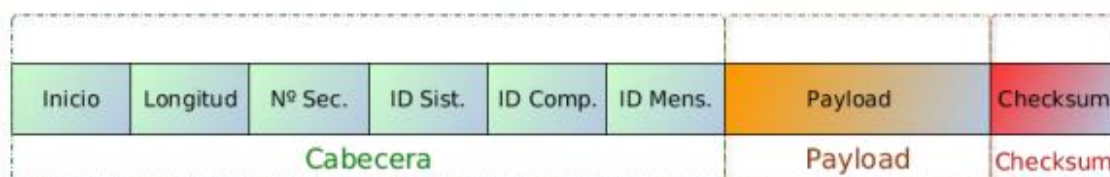


Fig. 4.2 Estructura paquet MAVLink

Cada camp vol dir el següent:

- Inici: denota l'inici de la transmissió
- Longitud: diu la longitud de bytes del Payload. Va de 0 a 255.
- N° seqüència: mostra el número de paquet, serveix per saber si s'ha perdut algun paquet pel camí.
- ID sistema: identifica el sistema. Serveix per a xarxes on hi ha més d'un sistema.
- ID component: identifica el component que envia el paquet. Permet identificar diferents components dins d'un mateix sistema.
- ID missatge: identifica el tipus de missatge.
- Payload: conté la informació a transmetre.
- Checksum: serveix per control d'errors

Un paquet MAVLink v1.0 pot tenir una longitud mínima de 6 bytes, i una longitud màxima de 263 bytes.

A més de tenir en compte els llenguatges de programació que suporta MAVLink, també hem tingut en compte que suportés el nostre controlador de vol i el GCS. És per això que utilitzem aquest protocol, perquè suporta el llenguatge Python, el controlador de vol Pixhawk 2.4.8 i l'estació de control a terra Mission Planner.

#### **4.2.3.2. Dronekit**

Dronekit és una API (*Application Programming Interface*) que permet crear aplicacions Python que es comunica amb vehicles a través de protocol MAVLink. En el nostre cas servirà per a la connexió amb un dron. Gràcies a Dronekit podem accedir de forma molt fàcil a la telemetria del dron, al seu estat i als paràmetres. A més també permet la gestió i el control directe dels moviments i les operacions del dron (veure [86]).

El gran avantatge de Dronekit, és que està pensada en la comunitat fet que fa que hi hagi una gran quantitat de documentació disponible i un gran nombre d'exemples desenvolupats per l'empresa que ha desenvolupat Dronekit com pels mateixos usuaris de l'API. Això fa que escollim aquesta API abans que altres APIs que estan centrades a desenvolupar les mateixes funcions.

De la mateixa forma que passa amb la llibreria Intel RealSense explicat en el punt 4.2.2, aquesta API ha d'estar instal·lada en la Raspberry Pi perquè el projecte que he dissenyat funcioni de forma correcta.

### **4.3. Funcionament del sistema de navegació**

Aquest apartat engloba tota la lògica i els procediments que s'han utilitzat per tal de desenvolupar el sistema de navegació indoor. Com hem explicat abans el principal protagonista del sistema de navegació indoor són les càmeres Intel

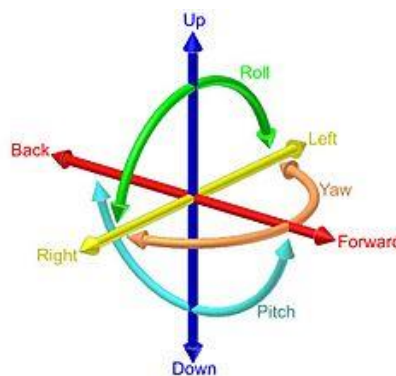
RealSense, ja que les dades que utilitzem com a referència pel sistema provenen d'elles. Pel sistema utilitzarem dues càmeres, la Intel Realsense T265 i la D435. La càmera T265 és la càmera que utilitzarem per al posicionament i la navegació, mentre que la càmera D435 la utilitzarem per aportar seguretat al dron, és a dir, detectarem obstacles per evitar col·lisions.

El primer pas per a desenvolupar el sistema de navegació indoor és saber quines són les variables de sortida que necessitem que vinguin de la càmera T265, i saber quines són les variables d'entrada que necessita el nostre dron per saber posicionar-se i desplaçar-se de forma equilibrada. Per tant, per a entendre el desenvolupament lògic del sistema, aquest apartat es dividirà en tres subapartats. El primer apartat tractarà sobre les dades que obtenim de la càmera T265 i la D435, i el segon apartat parlarà sobre les dades d'entrada que necessita el nostre dron. Finalment hi haurà un tercer apartat on s'explicarà la lògica i el funcionament del sistema de navegació que dissenyem.

#### 4.3.1. Obtenció dades provinents de les càmeres

En aquest bloc veurem com interpretar les dades que provenen de la càmera T265 i la D435. Començarem amb la T265, ja que és la càmera que utilitzem per al sistema de navegació en si, i la D435 la utilitzem com a seguretat.

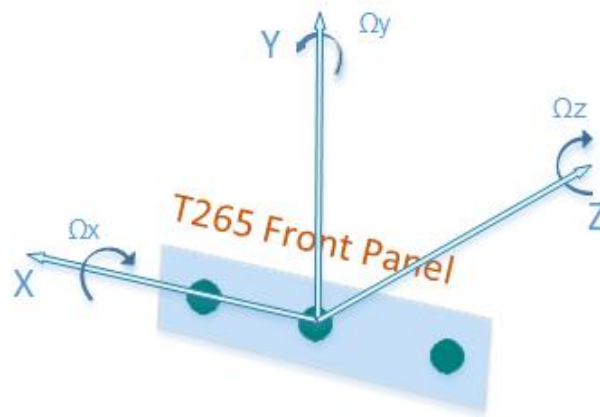
Com hem explicat abans la càmera T265 utilitza el mètode V-SLAM, que és un mètode que crea un mapejat d'un lloc desconegut i va actualitzant la nostra posició dins d'aquest mapejat de forma precisa. D'aquesta forma la càmera és capaç de posicionar-se dins de qualsevol espai, inclús aquells en els que no arribi senyal GPS. La càmera té sis graus de llibertat, igual que els graus de llibertat que té el dron, per tant, les dades que volem aconseguir de la càmera T265 són les que ens defineixen l'actitud de la càmera en aquests sis graus de llibertat. Dels sis graus de llibertat, tres corresponen a moviments de translació, és a dir, moviments respecte als tres eixos de coordenades (x, y, z) i els altres tres graus es corresponen a tres moviments de rotació, que són les rotacions respecte als tres eixos de coordenades (*pitch*, *roll* i *yaw*). Els sis graus de llibertat es poden veure en la Figura 4.3.



**Fig. 4.3** Sis graus de llibertat de la càmera T265

Com hem explicat abans, les variables que volem aconseguir de la càmera són  $x$ ,  $y$ ,  $z$ ,  $pitch$ ,  $roll$  i  $yaw$ . I les variables que obtenim de la càmera són el vector de posició ( $xyz$ ), el quaternió d'orientació ( $wxyz$ ), la velocitat lineal i els vectors d'acceleració, la velocitat angular i els vectors d'acceleració i també la confiança del seguiment (alt, mig, baix) (veure [87]). Per tant, com veiem no ens dóna els valors de  $pitch$ ,  $roll$  i  $yaw$ , però en canvi si ens dóna el vector de posició.

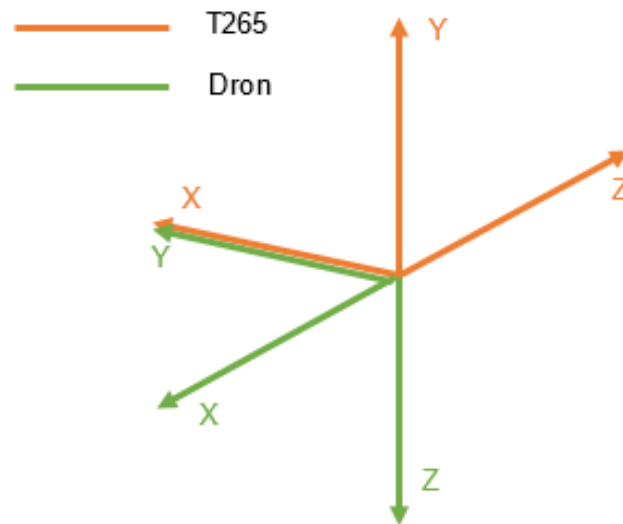
Respecte al vector de posició que proporciona la càmera tenim diferències importants amb el vector de posició que enviarem al dron. Les diferències entre els vectors de posició són el sistema de referència que utilitzen. La càmera T265 està referenciada en el sistema de referència que s'utilitza en la realitat virtual. El sistema de referència de la càmera T265 es pot veure en la Figura 4.4.



**Fig. 4.4** Sistema de referència Intel RealSense T265

De la Figura 4.4 podem observar que la X apunta cap a la dreta, la Y apunta cap a dalt i la Z apunta cap enrere. Un cop tenim definit el sistema de referència de la càmera l'hem de comparar amb el sistema de referència del dron per saber si hem d'aplicar alguna rotació per passar d'un sistema de referència a un altre. Segons la documentació del controlador de vol del dron, utilitza un sistema de referència FRD (de l'anglès "X Forward, Y Right and Z Down") per al sistema de referència local, és a dir, X apunta endavant, la Y apunta a la dreta i la Z apunta cap a baix. El sistema de referència del dron respecte al món és el NED (de l'anglès "X North, Y East and Z Down"). Per tant haurem de convertir el sistema de referència de la T265 amb el sistema de referència local del dron.

Per convertir el sistema de referència de la T265 cap al del dron mirarem la Figura 4.5. per veure les diferències entre els dos sistemes.



**Fig. 4.5** Sistema de referència de la càmera T265 comparat amb sistema de referència del dron

Com podem veure a la Figura 4.5, la conversió del sistema de referència de la càmera amb el del dron és molt senzill, per tant,  $x$ ,  $y$  i  $z$  quedarà de la següent forma:

$$\begin{aligned} x_{dron} &= -z_{T265} \\ y_{dron} &= x_{T265} \\ z_{dron} &= -y_{T265} \end{aligned} \quad (4.1)$$

Fent aquests canvis obtindrem el vector de posició del dron de forma correcta. Arribat aquest punt on ja tenim el vector de posició, tan sols falta aconseguir els angles de navegació, també nomenats angles de Tait-Bryan. Per aconseguir els angles no ho podem fer de forma directa, ja que la càmera no els proporciona, però sí que proporciona el quaternió d'orientació.

Els quaternions són una extensió dels nombres reals, semblant als nombres complexos. Es poden utilitzar per a representar l'orientació d'un objecte en un espai. L'ús del quaternió per a representar una rotació en comptes de l'ús d'angles d'Euler és degut a l'efecte "Gimbal Lock" i a l'eficiència i estabilitat. Aquest efecte fa que es perdi un grau de llibertat a causa de l'alineació de dos eixos de rotació. Un quaternió és considerat com  $\mathbb{R}$  - espai vectorial de quatre dimensions (veure [88] i [89]).

A partir de l'ús d'un quaternió podem trobar els angles de navegació que volem enviar al controlador de vol. Per poder crear un quaternió hem de crear una matriu de quaternió amb els paràmetres  $[wxyz]$  que obtenim de la càmera T265. Com que el sistema de referència de la càmera i el del dron no és el mateix, en crear la matriu de quaternió haurem de tenir en compte els canvis que hem trobat abans (equació 4.1) per passar d'un sistema de referència a un altre. Un cop

creem la matriu de quaternió amb els valors correctes podem extreure els angles de navegació amb el sistema de referència adequat, en aquest cas el del dron.

El treball d'extreure les dades de la càmera T265 ja s'ha acabat, ja que aconseguim treure els valors de les variables,  $[x\ y\ z\ pitch\ roll\ yaw]$ , que són les necessàries per a la navegació.

La càmera D435 que s'utilitza com a l'element de seguretat a l'hora de volar el dron per evitar col·lisions amb obstacles. Per evitar una col·lisió amb un objecte el dron necessita saber tan sols la distància a la qual es troba. La càmera D435 és una càmera de profunditat i com a tal, és capaç de calcular la distància de cadascun dels píxels que detecta la càmera. Per poder detectar un obstacle, recorrem tots els píxels que obté la càmera D435, i de tots els píxels mesurem la distància. A partir d'aquestes distàncies creem un histograma dels píxels. Aquest histograma recull la distribució de freqüències. Aleshores decidim el nombre de píxels per sota de la distància de seguretat (que decidim nosaltres) que servirà com a llindar per decidir si hi ha un obstacle o no. Si el nombre de píxels que tenen una distància inferior a la de seguretat és major al nivell llindar que marquem considerarem que hi ha un obstacle. Un cop decidim que hi ha un obstacle, a través de l'histograma calculem la distància mitjana a la qual es troba aquest obstacle i és aquest valor el que enviem al controlador de vol.

Un cop ja s'ha explicat les dades que obtenim provinents de les càmeres, en el següent apartat veurem la forma en què s'envien i com ho rep el controlador de vol del nostre dron.

#### 4.3.2. Enviament dades al controlador de vol

En aquest apartat, com ja hem dit, parlarem la forma en què enviem les dades obtingudes de les càmeres Intel RealSense al controlador de vol del nostre dron. Segons l'explicat a l'apartat anterior utilitzem la càmera T265 i la D435. La càmera T265 l'utilitzem per al posicionament i la navegació, i la D435 l'utilitzem per seguretat. De la càmera T265 les dades que obtenim són:  $[x\ y\ z\ pitch\ roll\ yaw]$ . De la càmera D435 obtenim la distància a la qual es troba un obstacle.

Tota la informació que obtenim amb les càmeres la passem al controlador de vol i alhora a l'estació de control a terra, en el nostre cas el Mission Planner. La comunicació entre el controlador de vol i l'estació de control es fa a través de missatges MAVLink, tal com hem explicat en apartats anteriors. Missatges MAVLink n'hi ha molts però s'han de seleccionar dos, un per l'enviament de la posició, i l'altre per l'enviament de la distància respecte a l'obstacle. Si mirem la llista de missatges MAVLink (veure [90]) que ja estan predefinits, els missatges que més s'ajusten als nostres requeriments són "VISION\_POSITION\_ESTIMATE" i "DISTANCE\_SENSOR".

El missatge "VISION\_POSITION\_ESTIMATE" és el missatge que utilitzarem per a transferir les dades obtingudes de la càmera T265. Segons el protocol MAVLink



aquest missatge serveix per transmetre la posició global i l'actitud del dron obtingut a través d'una font de visió, exactament el nostre cas (veure [90]). Els paràmetres que s'han de passar per tal d'enviar aquest missatge de forma correcta són els següents:

$$\left\{ \begin{array}{l} temps [ms] \\ x [m] \\ y [m] \\ z [m] \\ pitch [rad] \\ roll [rad] \\ yaw [rad] \end{array} \right.$$

Com veiem, els paràmetres que s'han d'enviar amb aquest missatge són els mateixos paràmetres que hem extret de la càmera T265. L'únic paràmetre que s'ha d'afegir que no es pot treure de la càmera és el temps, que concretament és el temps UNIX, que és la quantitat de segons que han passat des de la mitjanit UTC de l'1 de gener de 1970.

El missatge "DISTANCE\_SENSOR" és el missatge que utilitzem per a transferir les dades obtingudes per la càmera de profunditat (veure [91]). Els paràmetres que s'han de passar per aquest missatge MAVLink són:

$$\left\{ \begin{array}{l} temps [ms] \\ dist_{min} [cm] \\ dist_{max} [cm] \\ dist_{obstacle} [cm] \end{array} \right.$$

El temps és el mateix que en el cas de l'altre missatge, és a dir, el temps UNIX. Els altres paràmetres es refereixen a la càmera. La distància mínima amb què treballa la D435 (aproximadament uns 20 cm), la distància màxima (aproximadament 10 m) i la distància amb l'obstacle, que és la distància que nosaltres calculem.

Un cop li passem aquests dos missatges MAVLink del controlador de vol a l'estació de control de terra hem de modificar alguns paràmetres a l'estació de terra perquè canviï la forma de posicionar-se i navegar del dron. El procés de determinar la posició i orientació del dron a través de l'anàlisi d'imatges, com és el nostre cas, es diu odometria visual (veure [92]). Per defecte el controlador de vol Pixhawk està configurat per a navegar i posicionar-se utilitzant la informació extreta pels giroscopis, compàs, baròmetre i el senyal GPS. Com que en el nostre cas no tindrem senyal GPS hem de configurar el dron perquè agafi la informació que prové dels missatges MAVLink que li enviem. Perquè el dron no intenti navegar a través del senyal GPS hem d'utilitzar un dels filtres de Kalman que té configurat el controlador de vol. Els filtres de Kalman poden ser configurats per navegar a través d'odometria visual.

Un filtre de Kalman és un algorisme que es basa en el model d'espai d'estats d'un sistema per a estimar l'estat futur i la sortida futura, realitza un filtratge òptim del senyal de sortida i elimina el soroll. Les equacions que implementen un filtre de

Kalman només tenen dependència d'una mostra anterior i una mostra actual. Les equacions implementen un estimador – corrector, perquè minimitza l'error estimat de la covariància. A partir del filtre de Kalman, sorgeix el filtre de Kalman estès, que és un filtre de Kalman que es pot utilitzar en sistemes no lineals.

En el cas del nostre controlador de vol podem utilitzar tres filtres de Kalman estesos. En el nostre cas utilitzarem el segon, denominat a l'estació de control amb l'abreviació "EKF2" (veure [93]). Utilitzarem el segon, ja que el primer és molt antic i ja no s'acostuma a utilitzar, i el tercer encara estan treballant en ell i no recomanen treballar-hi excepte en determinats casos.

L'EKF2 permet subscriure's a temes d'odometria visual, a través del missatge MAVLink "VISION\_POSITION\_ESTIMATE" (veure [94]) i per tant permet processar les dades provinents de la càmera T265. Tot i que l'EKF2 permet utilitzar les dades que volem s'ha de configurar el controlador de vol per tal que faci cas a aquestes dades, per configurar els paràmetres del controlador de vol s'han de fixar a través de la nostra estació de control de terra. Els paràmetres que hem configurat per la utilització de l'EKF2 amb odometria visual són:

- GPS\_TYPE = 0 -> no utilitzarem GPS
- COMPASS\_USE = 0 -> no utilitzarem cap compàs per la rotació respecte a l'angle yaw
- AHRS\_EKF\_TYPE = 2 -> per definir que utilitzem l'EKF2
- EKF2\_ENABLE = 1 -> per activar l'EKF2
- EK2\_GPS\_TYPE = 3 -> no utilitzar el GPS en l'EKF2

En aquest subapartat hem vist quins són els missatges MAVLink que utilitzem per transmetre les dades obtingudes de les càmeres al controlador de vol i l'estació de control de terra.

### 4.3.3. Funcionament i lògica del sistema

El funcionament del sistema és molt senzill, partim de la base que tenim tres dispositius externs al nostre ordinador de bord que és l'encarregat d'executar el sistema. Els tres elements externs són les dues càmeres Realsense (la T265 i la D435) i el controlador de vol Pixhawk. Per tant, el primer pas del nostre sistema és iniciar la comunicació amb aquests dispositius. Amb el controlador de vol ens connectem a través dels ports UART, i amb les càmeres a través d'USB 2.0. Un cop establim la connexió hem de fixar uns paràmetres de configuració, per la connexió amb el controlador de vol hem de fixar el rati de baudis, que és el nombre d'unitats de senyal per segon. Per a la connexió amb les càmeres hem de fixar que volem obtenir, en el nostre cas informació sobre la posició i la distància, i permetre la gravació de les càmeres.

Després d'inicialitzar les connexions amb els dispositius externs donem pas a l'execució d'un bucle. Aquest bucle executarà ordres de manera seqüencial i infinita, fins que nosaltres vulguem. A cada iteració el sistema executa les següents ordres:

- Obtindre les dades de posició i rotació provinents de la càmera T265
- Obtindre les dades de profunditat provinents de la càmera D435
- Transformació de les dades de la càmera T265 al sistema de referència del controlador de vol
- Crear la matriu de quaternió a partir de les dades de rotació
- Extreure els angles Tait Bryan en radians (*roll*, *pitch* i *yaw*) de la matriu de quaternió
- Analitzar les dades de profunditat i calcular la distància a un objecte si aquest es troba per sota de la distància de seguretat
- Enviar a través de MAVLink els missatges "VISION POSITION ESTIMATE" i "DISTANCE SENSOR".

Durant l'execució del nostre sistema, a cada iteració obtenim les dades de tan sols un fotograma de cada càmera. La duració d'una iteració és molt petita i per tant per cada segon que passa, obtenim la informació de més d'un fotograma.

Per acabar d'entendre com funciona el sistema que hem dissenyat i programat s'ha de veure el codi desenvolupat en l'apartat 4 dels annexos. Juntament amb el codi hi ha un conjunt de comentaris que ajuden a l'enteniment d'aquest.

## 4.4. Resultats

Aquest apartat es basa en la presentació dels resultats en conseqüència d'haver utilitzat el sistema de navegació indoor que hem dissenyat en un dron. En el sistema que hem programat, a més d'haver creat el sistema de navegació indoor també hem afegit un parell de funcions per tal de poder extreure els resultats de cada prova de vol. Aquestes funcions extres es dediquen principalment a l'enregistrament en vídeo de les imatges que capturen les dues càmeres i guardar en un full de càlcul les dades de la trajectòria que ha seguit el dron.

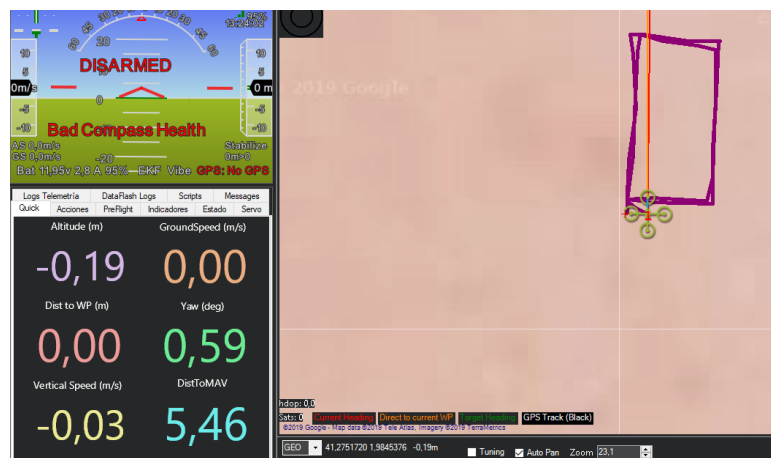
El sistema que hem dissenyat és molt complet, és capaç de connectar-se amb les dues càmeres alhora i amb el controlador de vol, és capaç d'obtenir les dades de posició per a crear un sistema de navegació indoor i també capaç de detectar objectes i calcular la distància fins aquests. A part de tot això també és capaç de gravar les imatges i guardar les dades de navegació. Tot això junt, fa que es necessiti una capacitat d'execució molt gran. En el nostre cas l'ordinador que s'encarrega de l'execució del sistema és una Raspberry Pi 3B amb tan sols 1 GB de RAM. La poca potència de la Raspberry Pi provoca que no es puguin utilitzar totes les funcions dissenyades alhora. Una de les funcions que hem de prescindir en executar el sistema en una Raspberry Pi és la de gravació d'imatges. Això és degut principalment a dues raons, una pel rendiment que suposa, i l'altre per la capacitat que ocupen les gravacions, ja que són de l'ordre de centenars de Mb. A més durant l'execució del programa a causa de la limitació dels ports USB de la Raspberry Pi, ja que són 2.0 en comptes de 3.0 l'obtenció d'imatges de la càmera de profunditat D435 és molt limitada, aquest fet fa que el sistema s'alenteixi fins al punt que acaba per aturar-se tota la Raspberry Pi.

A causa dels dos inconvenients que acabem de comentar i tot que el sistema té la possibilitat d'executar unes funcions més completes, les proves del sistema per a l'obtenció de resultats només s'han efectuat amb la càmera de seguiment T265. Això fa que les funcionalitats per gravar des de les càmeres i la de detectar objectes no s'hagin executat. Tot i això, el sistema restant compleix amb l'objectiu fixat, és a dir, el de navegar en un espai indoor.

Per demostrar que el sistema de navegació indoor funciona de forma correcta es comentaran els resultats de tres proves realitzades durant l'execució del sistema. La primera prova és una prova "en parat", aquesta prova serveix per veure que l'estació de control de terra rep la informació de forma correcta. A més es compararà la trajectòria que es mostra en l'estació de control de terra amb la trajectòria que emmagatzema el sistema que hem dissenyat. La segona prova serà una demostració del vol del dron durant l'execució del sistema. Per últim en la tercera prova farem que el dron executi ordres de desplaçament, és a dir, li direm que es mogui fins a una posició determinada. A través de l'execució d'aquesta prova veurem quina és l'exactitud del sistema de navegació indoor dissenyat. Totes les proves de vol es gravaran per tal de comprovar els resultats que es mostraran.

#### 4.4.1. Prova de terra

La prova de terra és molt important, ja que, és la prova prèvia a les proves de vol. En la prova de terra el que hem fet, és realitzar una trajectòria en forma de rectangle per tal de veure que el sistema de referència és correcte i a l'estació de control de terra li arriben les dades tal com nosaltres volem. Per saber si el sistema és correcte, comparem la trajectòria que es dibuixa en el Mission Planner (Figura 4.6) amb la trajectòria que dibuixem a partir de les dades de la càmera T265 (Figura 4.7).



**Fig. 4.6** Visualització de la trajectòria del dron durant la prova en terra vist des de l'estació de control de terra (Mission Planner)



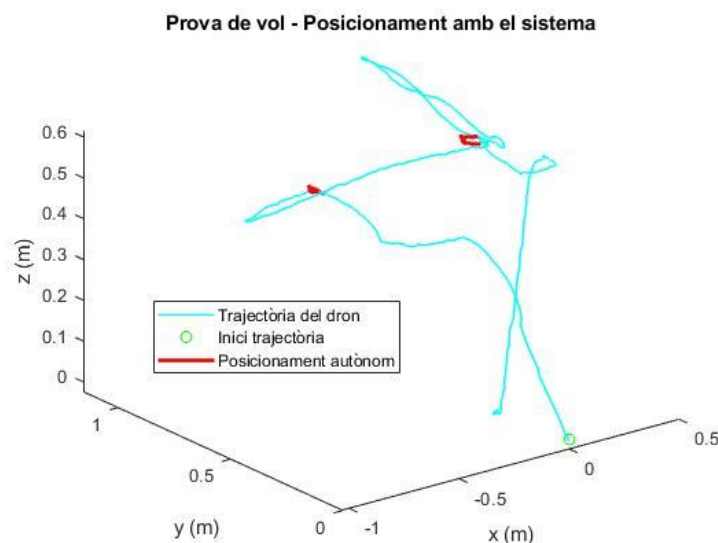
#### 4.4.2. Prova de posicionament

La prova de posicionament és la primera prova de vol realitzada per veure si el sistema de navegació actua de forma correcta i eficient. La realització d'aquesta prova s'ha dut a terme en un espai interior on no arriba el senyal GPS. Tot i que el senyal GPS no estava disponible s'ha desactivat l'obtenció de senyal GPS. La missió durant aquesta prova de vol és enlairar el dron, activar el sistema de navegació dissenyat, i després d'haver enviat un parell d'ordres al dron a través del ràdio control, deixar d'enviar ordres i veure com el dron és capaç de mantenir la posició.

Per acabar d'entendre els avenços és important recordar la prova de vol feta en un espai indoor sense senyal GPS i sense el sistema de navegació (veure punt 3.3.). Respecte al vol sense sistema de navegació vam veure com el dron un cop deixàvem d'enviar ordres es seguia movent de forma lleugera, ja que no era capaç de saber posicionar-se.

Per veure els resultats del vol s'ha de veure el vídeo adjuntat en els **annexos en el punt 3.3.** Respecte a la prova de vol observem com un cop deixem d'enviar ordres al dron des del ràdio control el dron és capaç de mantenir la posició de forma quasi perfecta, sense que s'observi cap moviment aparent. Aquest comportament és el mateix comportament que tenia el dron durant la prova de vol en un espai exterior amb el senyal GPS.

En la figura 4.9 podem veure la trajectòria del dron durant aquesta prova de vol. Com veiem en la trajectòria en 3D els moments on el dron ha mantingut la posició s'han pintat de color vermell per a diferenciar-los de la resta. Si observem les parts de les trajectòries de color vermell, podem apreciar que el moviment del dron durant el posicionament ha sigut gairebé inexistent.



**Fig. 4.9** Trajectòria del dron durant la prova de vol per comprovar el posicionament del dron amb el sistema de navegació dissenyat

Per tant, a partir d'aquesta prova de vol considerem que el sistema de navegació indoor funciona de forma correcta, ja que és capaç de mantenir al dron de forma estabilitzada a l'aire sense que aquest es mogui.

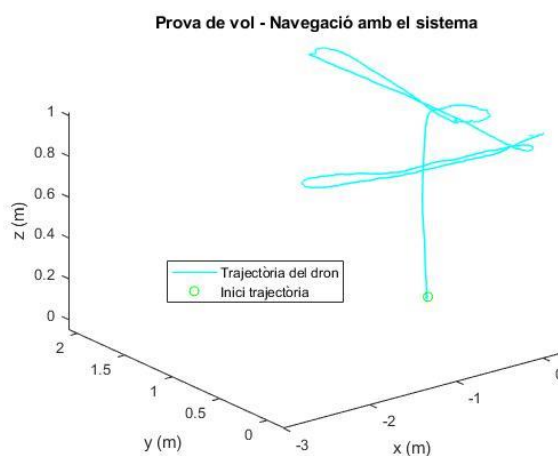
#### 4.4.3. Prova de navegació

Aquesta prova de vol és la més important de totes i la definitiva. Fins ara, en un espai interior on no arriba el senyal GPS el dron no és capaç de posicionar-se. Com que no es posiciona és impossible que li diguem que vagi d'un punt A a un punt B. Amb aquesta prova de vol el que volem provar és l'enviament d'instruccions de navegació al dron. Com a resultat de la prova el que haurem d'obtenir és el moviment del dron en les direccions que nosaltres li marquem a través del mapa del Mission Planner.

De la forma en la qual hem dissenyat el sistema, en inicialitzar-lo, el dron es posiciona en el mapa a l'esplanada de sorra de l'EETAC. D'aquesta forma quan li enviem les ordres de navegació, aquestes ordres hauran de ser respecte al punt d'origen fixat.

Per veure de forma clara els resultats d'aquesta prova de vol és necessari veure el vídeo adjuntat en el **punt 3.4 dels annexos**.

Respecte a la prova de navegació podem veure com el dron efectua els moviments que li manem des de l'estació de control a terra sense cap problema i en tot moment de forma estabilitzada. Quan el dron acaba d'efectuar el moviment es manté en la posició que ha quedat sense que s'aprecii cap classe de moviment fins que li executem el següent ordre. A partir de les dades que emmagatzemem durant el vol podem comparar les imatges del punt 3.4 dels annexos amb la trajectòria emmagatzemada. Aquesta trajectòria es pot veure en la figura 4.10.



**Fig. 4.10** Trajectòria del dron durant la prova de vol per comprovar navegabilitat amb el sistema de navegació dissenyat

## CONCLUSIONS

Aquest treball de fi de grau té com a finalitat la creació d'un sistema de navegació indoor. L'elecció de dissenyar un sistema de navegació indoor és motivat per la manca de senyal GPS als interiors d'edificis, sota terra, fàbriques, etc. El GPS és el sistema de navegació més utilitzat en el món, per tant, en tots els espais on no arriba senyal GPS no és possible navegar de per sí, necessitem la utilització d'un altre sistema de navegació. A més de dissenyar un sistema de navegació, un dels objectius, és que aquest sigui de baix cost, ja que d'aquesta manera és possible portar-ho a la realitat i fer proves amb ell. Per utilitzar el sistema de navegació hem construït un quadricòpter.

Per poder desenvolupar un sistema de navegació és molt important escollir bé la font d'on s'obtenen les dades. Per tal de fer una bona selecció hem fet un estudi de l'art de les tecnologies que s'estan utilitzant per a sistemes de navegació. A partir d'aquest estudi hem seleccionat com a font d'informació l'òptica. Hem escollit dues càmeres de la família Intel RealSense, una càmera de seguiment, la T265 i una càmera de profunditat, la D435. Per al desenvolupament del sistema de navegació la càmera de seguiment s'utilitza per a la navegació en si, i la càmera de profunditat com a seguretat, per a detectar obstacles. Amb la utilització d'aquestes dues càmeres el sistema és "ego-motion", considerem que és un punt positiu del sistema de navegació, ja que només necessitarà una única instal·lació en el dron i es podrà utilitzar en qualsevol espai.

Paral·lelament al disseny del sistema de navegació s'ha construït un quadricòpter, per poder integrar el sistema de navegació i anar comprovant tot el desenvolupament tecnològic. Per seleccionar les peces del dron, s'ha utilitzat el criteri del baix cost. Amb aquest criteri el dron construït ha costat un total de 245 €.

Amb el dron construït, tot i tenir problemes per aconseguir fer-lo funcionar de forma correcta s'han dut a terme les primeres proves de vol. Com a resultat de les primeres proves de vol hem vist la diferència entre volar un dron sota condicions de senyal GPS i volar un dron sense senyal GPS. En les proves de vol veiem que el posicionament del dron sense senyal GPS és dolent, ja que no aconsegueix mantenir la posició per si sol. Amb aquestes proves de vol, hem comprovat la necessitat de desenvolupar un sistema de navegació per espais on no arriba el senyal GPS.

Respecte al desenvolupament del sistema de navegació indoor ens ha sorgit algun problema. El primer problema que va sorgir va ser la utilització de quaternions per a l'obtenció de dades de rotació de la càmera de seguiment. Aquest terme matemàtic no l'havíem vist mai i ens hem hagut que posar al dia per saber com treballar amb ell. Gràcies al llenguatge de programació utilitzat, el Python, i una de les seves llibreries ha sigut factible convertir les dades del quaternió en les dades que ens interessaven.

El segon problema que ens va sorgir va ser durant les proves de funcionament del sistema. Per a poder utilitzar el sistema de navegació, és necessari que



aquest funcioni en l'ordinador de bord del dron. Hem seleccionat una Raspberry Pi 3B amb 1 Gb de RAM i ports USB 2.0, ja que era un ordinador de bord molt assequible. A causa de la potència limitada de la Raspberry Pi i la utilització de ports USB 2.0 en comptes de ports USB 3.0 ha fet que no sigui possible utilitzar totes les funcionalitats extres amb les quals havíem dissenyat el sistema de navegació. Les funcionalitats que no hem pogut utilitzar són la de detectar objectes amb les imatges provinents de la càmera de profunditat D435 i la de gravar les imatges que veien d'ambdues càmeres. Tot i això, les funcionalitats restants han sigut les essencials per fer funcionar el sistema de navegació de forma correcta.

Respecte al cost del sistema, el cost total del sistema és de 430 €, 200 € de cada càmera i 30 € de la Raspberry Pi. Com que no hem pogut utilitzar les dues càmeres durant les proves de vol, i només hem utilitzat la càmera de seguiment T265, el cost del sistema amb el qual hem efectuat les proves de vol és de 230 €.

Quant a les proves de vol amb el sistema de navegació hem vist la gran capacitat que té el dron de posicionar-se amb el sistema. Com hem vist, el dron es manté de forma quasi exacta en la seva posició sense observar cap moviment aparent. Respecte a la prova de vol de navegabilitat hem vist com el dron és capaç d'entendre les nostres ordres de desplaçar-se d'un punt A a un punt B. A partir dels resultats considerem que els objectius que ens hem marcat en el treball han quedat més que assolits. A més, un dels paràmetres amb els quals s'ha dissenyat el sistema de navegació és que aquest es pugui utilitzar en qualsevol classe de vehicles.

Com a resultat del treball i durant el desenvolupament d'aquest han sorgit aspectes en els quals es podria treballar de forma àmplia en futurs treballs. Aquests temes podrien ser com afecta canvis en la intensitat de la llum en el sistema de navegació, el comportament del sistema si hi ha vent, el comportament del sistema de navegació en un espai molt gran, com podria ser una fàbrica, i després d'un temps llarg de funcionament. I per últim veure el funcionament en un vol amb totes les funcionalitats que no hem sigut capaços d'utilitzar, la detecció d'objectes i l'enregistrament d'imatges.

## BIBLIOGRAFIA

- [1] PricewaterhouseCoopers, (2016). "Clarity from above" .
- [2] GDP Consultoria, "El mercado de los drones, negocio con futuro en España" <http://gdpconsultoria.com/los-drones-negocio-de-futuro-en-espana/>
- [3] Ministeri de Foment del Govern Espanyol, (2018). "Plan Estratégico para el Desarrollo del Sector civil de los Drones en España (2018-2021)"
- [4] Zenit drones, "U-Space: El gran reto europeo para los drones". <https://zenitdrones.com/u-space-gran-reto-europeo-para-los-drones/>
- [5] Dargie W., Poellabauer C., "Localization" Cap. 10 en *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons, pp 249-266. (2010)
- [6] Unigis, "Posicionamiento indoor". <https://www.unigis.es/posicionamiento-indoor/>
- [7] Díaz-Ambrona Tabernilla, L., "Sistema de localización en interiores". Treball Fi de Grau en Universidad Politécnica de Madrid.
- [8] Seco Granja F., (2007/08). "Sistemas de localización en interiores basados en radiofrecuencia", Tema 3.3 en "Sistemas Inteligentes". Universidad de Alcalá.
- [9] NetSpot, "Qué es RSSI y su relación con una red Wi-Fi". <https://www.netspotapp.com/es/what-is-rssi-level.html>
- [10] Subedi S. i Pyun J-Y., (2017) "Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons".
- [11] Red Inalámbrica "WIFI", "Definición de WIFI". <https://redwifi.wordpress.com/definicion-de-wifi/>
- [12] Shah D. i Shah K., (2012). "Basic of Wi-Fi based positioning System".
- [13] Leverage, "WiFi for Indoor Positioning". <https://www.leverage.com/blogpost/wifi-indoor-positioning>
- [14] Infsoft, "Indoor Positioning, Tracking and Indoor Navigation with Wi-Fi". <https://www.infsoft.com/technology/sensors/wifi>
- [15] Núñez Sobrino, M<sup>a</sup>. A., "Sistema de localización en redes wi-fi basado en fingerprint". Treball Fi de Grau en Universidad Carlos III de Madrid.
- [16] ELT, "BLE (Bluetooth Low Energy)". <https://www.elt.es/ble-bluetooth-low-energy>
- [17] Locatify, "iBeacons or Ultra-Wideband for Indoor Proximity Services". <https://locatify.com/blog/ble-beacons-vs-ultra-wideband-for-indoor-gps-proximity-rtls/>
- [18] Iot for all, "Indoor Positioning with Bluetooth Low Energy (BLE)". <https://www.iotforall.com/indoor-positioning-bluetooth-low-energy-ble/>
- [19] Proximi, "How to do accurate indoor positioning with Bluetooth beacons?". <https://proximi.io/accurate-indoor-positioning-bluetooth-beacons/>
- [20] Aykaç M., Erçelebi E. i Baha Aldin N., (2017). "ZigBee-based indoor localization system with the personal dynamic positioning method and modified particle filter estimation"
- [21] Huircán, J.I., Muñoz, C., Young, H., Von Dossow, L., Bustos, J., Vivallo, G. i Toneatti, M., "ZigBee-based wireless sensor network localization for cattle monitoring in grazing fields", Volumen 74 en "Computers and Electronics in Agriculture", pp 258-264, (2010).
- [22] SearchDataCenter en Español, "ZigBee" <https://searchdatacenter.techtarget.com/es/definicion/ZigBee>

- [23] ZigBee Alliance, "Our membres". <https://www.zigbee.org/zigbeealliance/our-members/>
- [24] Efecto LED, "Qué es ZigBee?". <https://www.efectoled.com/blog/que-es-zigbee/>
- [25] El Androide Libre, "Todo sobre ZigBee, la tecnología ultrabarata para comunicación inalámbrica". <https://elandroidelibre.lespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>
- [26] Millán, R.J., (2004). "UWB (Ultra Wide-Band)".
- [27] Diaz, J., (2003). "Tecnología ultra-wideband (uwb). La revolución a corto alcance"
- [28] Lifewire, "What Does UWB Mean?". <https://www.lifewire.com/ultra-wide-band-817953>
- [29] Tendencias de la Ingeniería, "Desarrollan un radar para localizar personas desaparecidas bajo los escombros". [https://www.tendencias21.net/Desarrollan-un-radar-para-localizar-personas-desaparecidas-bajo-los-escombros\\_a1331.html](https://www.tendencias21.net/Desarrollan-un-radar-para-localizar-personas-desaparecidas-bajo-los-escombros_a1331.html)
- [30] Insoft, "UWB: Two Localization Techniques in Comparison". <https://www.insoft.com/blog-en/articleid/305/uwb-two-localization-techniques-in-comparison>
- [31] Abdulrahman, A., AbdulMalik, A., Mansour, A., Ahmad, A., Suheer, A., Mai, A., Hend S., A., (2016). "Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances".
- [32] Mautz, R., (2012). "Indoor positioning technologies".
- [33] Nuoplanet, "¿Qué es RFID?" <https://nuoplanet.com/blog/que-es-rfid/>
- [34] Dipolerfid, "Tecnología RFID" <https://www.dipolerfid.es/es/tecnologia-RFID>
- [35] Grupo FRACTALIA, "¿Para qué sirve la tecnología RFID?" <https://www.fractaliasystems.com/para-que-sirve-la-tecnologia-rfid/>
- [36] Universidad Internacional de Valencia, "Rfid: qué es y como funciona" <https://www.universidadviu.es/rfid-que-es/>
- [37] Dipolerfid, "Tag RFID". <https://www.dipolerfid.es/es/Tag-RFID>
- [38] Robots argentina, "Comunicación por infrarojos". [http://robots-argentina.com.ar/Comunicacion\\_IR.htm](http://robots-argentina.com.ar/Comunicacion_IR.htm)
- [39] Israel, A. i Dehy, R., (2004). "Infrared tracking system".
- [40] AirFinder, "How An Indoor Positioning System Works". <https://www.airfinder.com/blog/indoor-positioning-system>
- [41] Kirchner, N. i Furukawa, T., (2005). "Infrared Localisation for Indoor UAVs".
- [42] Raharijaona T., Mawonou, R., Nguyen, T., Colonnier, F., Boyron, M., Diperi, J. i Viollet, S., (2017). "Local Positioning System Using Flickering Infrared LEDs".
- [43] AT&T Laboratories, "The Bat Ultrasonic Location System". <https://www.cl.cam.ac.uk/research/dtg/attarchive/bat/>
- [44] M. I. T. Computer Science and Artificial Intelligence Laboratory, "The Cricket Indoor Location System". <http://cricket.csail.mit.edu/#acknowledgments>
- [45] Semantic Scholar, "DOLPHIN: an autonomous indoor positioning system in ubiquitous computing environment". <https://www.semanticscholar.org/paper/DOLPHIN%3A-an-autonomous-indoor-positioning-system-in-Fukuju-Minami/4d1618f52d0920e315877741184910f1a602aac1>
- [46] Leverage, "Indoor Positioning with Ultrasonic/Ultrasound". <https://www.leverage.com/blogpost/ultrasonic-indoor-positioning>

- [47] Qi, J. i Liu, G.P., (2017). "A Robust High-Accuracy Ultrasound Indoor Positioning System Based on a Wireless Sensor Network"
- [48] Holm, S., (2012). "Ultrasound positioning based on time-of-flight and signal strength".
- [49] Sakperrea, W., Adeyeye-Oshinb, M. i Mlitwac, N., (2017). "A state-of-the-art survey of indoor positioning and navigation systems and technologies".
- [50] Mautz, R. i Tilch, S., (2011). "Survey of Optical Indoor Positioning Systems"
- [51] Wang, Y., Peng C., Ravankar, A. i Ravankar, A., (2018). "A Single LiDAR-Based Feature Fusion Indoor Localization Algorithm".
- [52] Chen, Y., Liu, J., Jaakkola, A. i Chen, R., (2014). "Knowledge-based indoor positioning based on LiDAR aided multiple sensors system for UGVs"
- [53] IEEE Spectrum, "Sweep Is a \$250 LIDAR With Range of 40 Meters That Works Outdoors". <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/sweep-lidar-for-robots-and-drones>
- [54] DroneZon, "12 Top Lidar Sensors For UAVs, Lidar Drones And So Many Great Uses". <https://www.dronezon.com/learn-about-drones-quadcopters/best-lidar-sensors-for-drones-great-uses-for-lidar-sensors/>
- [55] Motorpasion, "Qué es un LIDAR, y cómo funciona el sensor más caro de los coches autónomos". <https://www.motorpasion.com/tecnologia/que-es-un-lidar-y-como-funciona-el-sistema-de-medicion-y-deteccion-de-objetos-mediante-laser>
- [56] Kumar, G., Kumar Patil, A., Patil, R., Sill Park, S., i Ho Chai, Y., (2017). "A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification"
- [57] Gibiansky, A., (2012). "Quadcopter Dynamics, Simulation, and Control".
- [58] David Schmidt, M., (2011). "Simulation and control of a quadroter unmanned aerial vehicle".
- [59] Teppo, L., (2011). "Modelling and control of quadcopter".
- [60] Tian, Y., (2019). "A mathematical modeling and development of control laws for Unmanned Aerial Vehicle (UAV)".
- [61] Solo drones baratos, "¿Qué tipus de drones existen? ¿Cuáles son sus precios y características?". <https://solodronesbaratos.com/tipos-de-drone/>
- [62] Getfpv, "All About Multirotor Drone FPV Frames". <https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-frame/>
- [63] Mydronelab, "Best quadcopter frames: building the drone from scratch". <https://www.mydronelab.com/accessories/quadcopter-frames.html>
- [64] Beginner flyer, "How to build a drone". <http://beginnerflyer.com/build-a-drone/>
- [65] Drone Casero, "APM VS NAZA, qué controladora es mejor". <http://dronecasero.blogspot.com/2015/07/apm-vs-naza-que-controladora-es-mejor.html>
- [66] My first drone, "Best flight controllers and why". <https://myfirstdrone.com/blog/best-flight-controllers>
- [67] Best drones for me!, "8 Best Drone Flight Controllers for Your Quadcopters in 2019". [http://bestdronesforme.com/best-drone-flight-controllers-quadcopters-board/#1\\_Flying\\_Style](http://bestdronesforme.com/best-drone-flight-controllers-quadcopters-board/#1_Flying_Style)
- [68] PromeTec, "Lo que hay que saber para elegir el controlador de vuelo (fc) en un multicóptero". <https://www.prometec.net/elegir-fc-para-dron/>
- [69] Mobus, "Motores Brushless para drones, todo lo que necesitas saber". <https://mobus.es/blog/motores-brushless-para-drones-todo-lo-que-necesitas-saber/>

- [70] Tutorial de drones, "Motor para drone". <https://www.tutorialdedrones.com/motor-para-drone/#quecomprar>
- [71] RobotShop, "How to make a Drone / UAV – Propulsion". <https://www.robotshop.com/community/tutorials/show/how-to-make-a-drone-uav-lesson-3-propulsion>
- [72] Drone trest, "How to choose the right motor for your multicopter drone". <https://www.dronetrest.com/t/how-to-choose-the-right-motor-for-your-multicopter-drone/568>
- [73] Rc Drone, "Simple formula about quadcopter weight calculator and thrust calculator". <http://www.rcdronegood.com/formula-quadcopter-weight-calculator-thrust/>
- [74] Drone nodes, "How to build a drone". <http://dronenodes.com/how-to-build-a-drone>
- [75] Drone center, "Construye tu dron". <http://dronecenter.blogspot.com/p/construye-tu-drone.html>
- [76] Mydronelab, "How to build a drone: construct your drone from scratch". <https://www.mydronelab.com/blog/how-to-build-a-drone.html>
- [77] Control dron, "Baterías de Drones". <https://www.controldron.com/tag/tipos-de-baterias-de-drones/>
- [78] Intel RealSense, "Intel® RealSense™ Depth Camera D435". <https://www.intelrealsense.com/depth-camera-d435/>
- [79] Intel RealSense, "Intel® RealSense™ Tracking Camera T265". <https://www.intelrealsense.com/tracking-camera-t265/>
- [80] Intel RealSense, "Robust Visual-Inertial Tracking from a Camera that Knows Where it's Going". <https://www.intelrealsense.com/visual-inertial-tracking-case-study/>
- [81] StackOverflow, "Why is Python Growing So Quickly?". <https://stackoverflow.blog/2017/09/14/python-growing-quickly/>
- [82] Genbeta, "Estas son las razones por las que muchos programadores están empezando a aprender Python". <https://www.genbeta.com/desarrollo/estas-razones-que-programadores-estan-empezando-a-aprender-python>
- [83] GitHub, "Librealsense". <https://github.com/IntelRealSense/librealsense>
- [84] Xdrones, "MAVLink: protocolo de comunicación para drones". <https://www.xdrones.es/mavlink/>
- [85] No es como lo cuentan, "Programando y desarrollando. – Protocolo MAVLink". <https://noescomolocuantan.wordpress.com/2014/10/18/programando-y-desarrollando-protocolo-mavlink/>
- [86] Dronekit. <http://dronekit.io/>
- [87] Intel RealSense, "How-to: Getting IMU data from D435i and T265". <https://www.intelrealsense.com/how-to-getting-imu-data-from-d435i-and-t265/>
- [88] Unity, "Rotación y Orientación en Unity". <https://docs.unity3d.com/es/2017.4/Manual/QuaternionAndEulerRotationsInUnity.html>
- [89] Arribas Raigadas, V., (2015). "Quaternions. Una rápida introducción".
- [90] MAVLink, "Vision position estimate". [https://mavlink.io/en/messages/common.html#VISION\\_POSITION\\_ESTIMATE](https://mavlink.io/en/messages/common.html#VISION_POSITION_ESTIMATE)
- [91] MAVLink, "Distance sensor". [https://mavlink.io/en/messages/common.html#DISTANCE\\_SENSOR](https://mavlink.io/en/messages/common.html#DISTANCE_SENSOR)
- [92] McGarey, P., (2016). "Visual Odometry (VO)".

- [93] Ardupilot, “EKF2 Estimation System”. <http://ardupilot.org/dev/docs/ekf2-estimation-system.html>
- [94] Ardupilot, “Using Vision or Motion Capture Systems for Position Estimation”. [https://dev.px4.io/v1.9.0/en/ros/external\\_position\\_estimation.html](https://dev.px4.io/v1.9.0/en/ros/external_position_estimation.html)

## ANNEXOS

### 1. Muntatge dron

El muntatge del dron és una de les parts més importants, ja que sense una instal·lació correcta de les seves peces volar-lo serà impossible. A més, en el cas d'aquest dron, on l'estructura que hem escollit és de dimensions ajustades tot haurà d'estar ben instal·lat per tal de tenir espai per a poder instal·lar el sistema de navegació que dissenyarem.

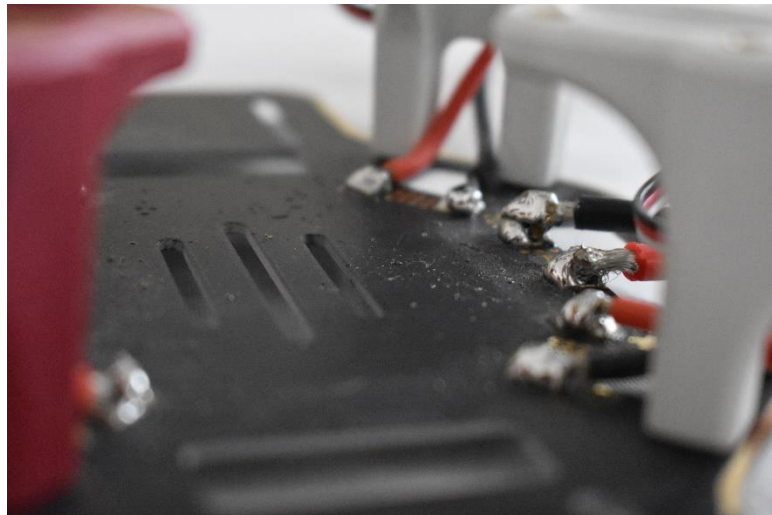
El punt central del nostre dron és la placa PDB (*Power Distribution Board*) que és una placa amb un circuit imprès on haurem de soldar diferents elements. Aquesta placa PDB és una de les parts de l'estructura del fuselatge, ja que aquesta placa està dissenyada per ser el punt de connexió dels diferents braços de la nostra estructura. La placa PDB ens permetrà fer un muntatge net i sense complicacions de tota l'electrònica del dron.

El primer pas és soldar els ESC a la placa PDB. Els variadors tenen un cable vermell i un negre i la placa PDB té dos punts situats en la posició on aniran els braços del dron. El cable vermell ha d'anar soldat en el punt positiu de la placa, i el cable negre ha d'anar soldat en el punt negatiu. D'aquesta forma quedaran soldats els quatre ESC en els punts on aniran units els braços amb la placa.

El següent pas és soldar un mòdul de potència a la placa distribuïdora. Normalment aquests mòduls de potència tenen un connector XT60 mascle i un altre femella. El connector mascle és el que va connectat a la bateria i el connector femella va connectat als variadors. Com que volem soldar-ho a la placa el que farem és tallar el connector XT60 femella, i els cables que queden són els que van soldats a la placa. D'aquesta forma amb una sola connexió alimentarem els quatre variadors gràcies a la placa PDB.

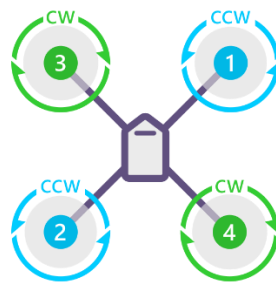
Un cop soldats els variadors i el mòdul de potència ja podem unir els braços i les potes pels aterratges. Amb aquests primers passos el muntatge ha de quedar de la següent forma:





**Fig. Annexos 1** Soldadura variadors a la placa

Un cop tenim els elements soldats i els braços i peus units al fuselatge, el següent pas és començar a unir els dispositius. Primer de tot cargolarem els motors als braços del fuselatge. Aquest pas és molt important, ja que els motors no poden anar en qualsevol posició. Els motors van per parelles. Tenim una parella de motors que gira en sentit horari i una altra parella de motors que gira en sentit antihorari. Per tant, la posició del motor dependrà del sentit de gir. Per saber en quina posició han d'anar col·locats ens fixarem en la figura 2.



**Fig. Annexos 2** Gir dels motors

Un cop ja tenim els motors cargolats de forma correcta hem d'unir els motors amb els variadors de cada braç del fuselatge. Als variadors els hi queden tres cables negres independents que tenen un connector femella i un altre cable que està format per tres cables petits de color vermell, negre i blanc. Els cables amb els connectors femella són els cables que s'uneixen amb els cables dels motors. És important veure la posició dels cables que surten dels ESCs, perquè la connexió amb els motors és important. En els motors que giren en sentit horari, és a dir els motors 3 i 4 segons l'anterior imatge, els cables dels variadors i motors van connectats normals, és a dir, els cables no es creuen. En canvi, en



el cas dels motors que giren en sentit antihorari la connexió dels cables que es troben en els extrems va creuada. D'aquesta manera els motors giraran en el sentit corresponent. El cable que sobra dels variadors és el que ha d'anar connectat al controlador de vol.

Un cop ja tenim tot el treball fet als braços del fuselatge tan sols queda col·locar el controlador de vol i després fer totes les connexions que necessita. Per fer el dron el més estable possible, necessitem que el controlador del vol estigui el més centrat possible, i que la fletxa que té dibuixat el controlador de vol apunti en la direcció frontal d'aquest. Ja només ens queda fer les connexions del *switch*, del *buzzer*, de l'emissora de ràdio, de la telemetria, del mòdul GPS, dels variadors ESC i del mòdul de potència. Per fer les connexions ens basarem la següent imatge del controlador de vol:

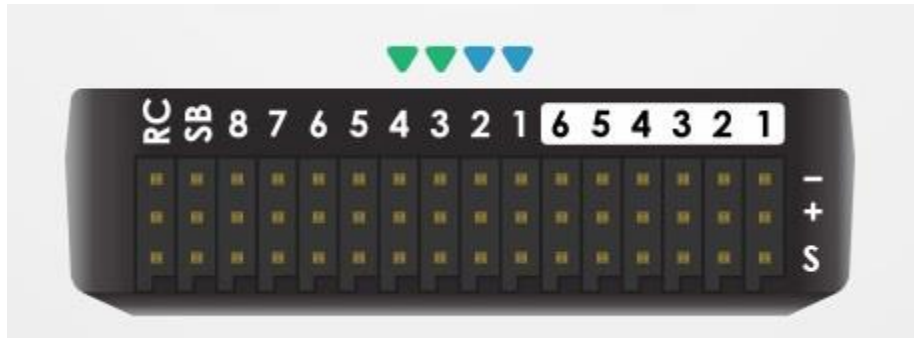


**Fig. Annexos 3** Controlador de vol Pixhawk

Les connexions aniran de la següent forma:

- *Switch*: connectat en el port "SWITCH"
- *Buzzer*: connectat en el port "BUZZER"
- Telemetria: connectat en el port "TELEM 1"
- Mòdul GPS: connectat en el port "GPS" i "I2C". El port I2C és pel compàs que incorpora el GPS
- Mòdul de potència: connectat en el port "POWER".

Per poder connectar els variadors i el receptor de ràdio utilitzarem els pins de sortida de la part frontal Pixhawk. Seguirem la figura 4 per realitzar les connexions.



**Fig. Annexos 4** Connexions dels motors

- Els ESCs aniran connectats segons el número de motor, ho podem veure a la figura 4.
- El receptor de ràdio es connecta en els pins "RC".

És important que tots els connectors estiguin connectats de tal forma que el cable negre dels dispositius que connectem estigui en la posició més elevada.

En els pins de sortida RC no va connectat directament el receptor de ràdio sinó que va connectat el codificador PPM. El codificador PPM connecta el receptor de ràdio de la nostra emissora de vol que té deu canals amb el controlador de vol. La seva funció és transformar els senyals PWM (*pulse width modulated*) que vénen de l'emissora de ràdio en un senyal PPM (*pulse position modulated*). Encara que la nostra emissora és de deu canals el codificador PPM només codifica vuit senyals PWM, per tant, només podrem utilitzar vuit canals.

Amb tots els dispositius connectats és important que la seva posició estigui en un lloc on no impedeixin el gir de les hèlixs i sobretot estiguin ben fixats.

Per últim col·loquem les hèlixs. La col·locació de les hèlixs ha de ser correcta, ja que només hi ha una configuració possible perquè el dron voli. Per saber la col·locació de les hèlixs ens fixem en el sentit de gir dels motors i en les mateixes hèlixs. Les hèlixs van per parelles, iguals que els motors, i també s'emparellen segons el sentit de gir que tindran. Per tant, les hèlixs aniran col·locades en creu. Per saber quines hèlixs han d'anar en els motors que giren en sentit horari i quines amb els de sentit antihorari ens fixarem amb el cantó de l'hèlix que està més elevat. Col·loquem les hèlixs de forma que el cantó elevat sigui el cantó que talla l'aire, el primer que entra en contacte amb l'aire en un gir, d'aquesta forma el moviment de les hèlixs generarà la sustentació que necessita el dron per poder volar.

Un cop ja tenim totes les connexions fetes, i tots els elements connectats l'aspecte del dron és el següent:



**Fig. Annexos 5** Dron construït

## 2. Configuració dron

Un cop ja tenim tots els elements connectats al controlador de vol i ben fixats en el fuselatge és hora de configurar el dron. La configuració és un pas vital, ja que si no es fa de manera correcta, el dron no volarà.

El primer de tot és elegir el *software* que s'utilitzarà per a l'estació terrestre. L'estació de terra és una aplicació que s'instal·la a l'ordinador que es trobarà a terra i que serà l'encarregat de comunicar-se amb el dron via telemetria sense fils. A més d'establir la connexió amb el dron quan aquest està volant també s'encarrega de tota la configuració del dron. El programa que hem escollit per a l'estació de terra és el Mission Planner. L'elecció d'aquest programa es deu bàsicament a tres factors. El software permet la connexió amb el controlador de vol Pixhawk, també permet modificar els paràmetres del controlador de vol, i per últim, la documentació del programa és molt extensa i molt detallada.

Un cop ja tenim el software que s'utilitzarà com a estació terra, el primer pas serà connectar aquest programa amb el controlador del nostre dron. Tenim dos possibles mètodes, a partir de la telemetria (sense fils) i a través d'USB (el controlador de vol té una sortida micro USB. Ara que ja tenim el controlador de vol connectat a l'ordinador i el Mission Planner obert haurem de descarregar el firmware que necessita el controlador de vol i instal·lar-ho. El firmware és el programa que porta instal·lat el controlador Pixhawk per tal de funcionar de manera correcta. El Mission Planner ens ofereix una gran quantitat de versions de firmware, que es van millorant amb el temps, hi ha versions beta i versions estables (versions que no estan en desenvolupament). A més també es pot carregar un firmware propi. En el nostre cas utilitzarem l'última versió estable de firmware per assegurar que el dron vola correctament.

Després d'instal·lar el firmware podem començar a la configuració del hardware del dron. Per a la configuració del hardware haurem de configurar el tipus de fuselatge, el compàs, la ràdio, l'acceleròmetre i els variadors.

- Configuració del fuselatge: primer de tot s'ha de configurar el tipus de fuselatge. El controlador de vol necessita saber primer de tot quina classe d'estructura és, és a dir, quants motors utilitza, si és un helicòpter o un dron, etc. En aquest cas seleccionem l'opció de Quadcopter. Un cop la classe ja està seleccionada, s'ha de seleccionar el tipus d'estructura, ja que segons la posició dels braços hi ha diferents tipus d'estructures. Com vam comentar en l'apartat de la selecció de l'estructura el tipus de *frame* que utilitzem és en X. Amb la classe i tipus d'estructura ja s'acaba aquest primer apartat.

- Configuració del compàs: la configuració del compàs és tan sols un calibratge d'aquest. El compàs és una unitat electrònica que permet saber al controlador de vol la seva orientació o rumb respecte al camp magnètic terrestre, i per tant cap a on es mou el dron. Per fer el calibratge del compàs, el Mission Planner ens dóna diferents opcions però totes elles molt semblants. La primera opció es diu *Onboard Calibration* (calibratge a bord) es realitza col·locant el dron en diferents posicions, tal com va dient el Mission Planner. L'altra opció es diu *Live Calibration* (calibratge en viu) i es tracta de girar el dron sobre els tres eixos.

- Calibratge del ràdio control: la configuració del ràdio control està dividida en dos passos, i aquest és el primer. El calibratge del ràdio control es basa a veure quins són els límits del nostre ràdio control. Per a realitzar el calibratge haurem d'encendre el transmissor de ràdio i moure tots els *sticks* i les palanques fins al límit del seu recorregut. Amb el calibratge el controlador de vol coneixerà els valors màxims i mínims del transmissor de ràdio. Un cop acabat el calibratge el Mission Planner ens donarà una llista amb els valors del calibratge per a cada canal. Normalment els valors mínims es troben sobre 1100 i els valors màxims sobre 1900.

- Calibratge acceleròmetre: l'acceleròmetre és la unitat electrònica que permet saber les actuacions del dron i per tant coneix la posició actual relativa a la posició immediatament anterior. És a dir, l'acceleròmetre és la unitat de mesura inercial del nostre dron. Per al calibratge d'aquest, haurem de col·locar el dron en les mateixes posicions que en el calibratge del compàs *Onboard Calibration*.

- Configuració del ràdio control: aquest és el segon pas de la configuració del ràdio control. El controlador de vol té l'opció de volar sota diferents modes, aquests modes poden ser mantenir l'altura, estabilitzat, tornar a casa, etc. Per a poder seleccionar cada mode és important que el controlador de vol estigui ben sincronitzat amb el transmissor de ràdio control. Aquesta sincronització es basa a seleccionar els mateixos modes de vol tant en el controlador com en el transmissor. Per fer-ho el Mission Planner ens dóna un interval de senyal PWM per a cada mode. Aquest interval serà el que haurem de posar en el nostre transmissor per a cada mode de vol. Un cop tinguem configurat tant en el transmissor com en el controlador hem de comprovar que el controlador detecta els canvis en el mode de vol. Per provar-ho canviarem de mode de vol en el transmissor i

observarem si el Mission Planner detecta un canvi en el mode de vol. Si el detecta, voldrà dir que la configuració està ben feta.

- Configuració dels variadors: aquest és l'últim pas en la configuració del hardware del dron. El calibratge dels variadors servirà perquè el controlador de vol conegui els valors màxims i mínims de PWM que el controlador enviarà. Per a poder realitzar el calibratge hem de seguir els següents passos:

1. Encendre el transmissor de ràdio amb la palanca de potència al màxim
2. Connectar la bateria al dron
3. Desconnectar la bateria
4. Pulsar el switch de seguretat del Pixhawk
5. Esperar que el dron emeti un soroll (entrem al mode calibratge d'ESC)
6. Baixar la palanca de potència del transmissor de ràdio al mínim
7. El dron emetrà un soroll llarg si el calibratge s'ha fet de forma correcta

En finalitzar tots els passos de configuració el dron ja està llest per a realitzar el primer vol.

### **3. Vídeos proves de vol**

#### **3.1. Amb senyal GPS**

En el següent enllaç es pot trobar el vídeo amb la primera prova de vol. Com es podrà veure el dron es troba en un espai exterior, on rep el senyal GPS. Enlairem el dron i el deixem a l'aire. Sense tocar el ràdio control el dron, gràcies al senyal GPS intenta mantenir la seva posició. Com es veu en el vídeo, l'efecte de l'aire fa que el dron hagi de lluitar contra aquest per mantenir la posició.

Enllaç al vídeo: <https://youtu.be/eqdp73AZAuc>

#### **3.2. Sense senyal GPS**

En el següent enllaç es pot veure el vídeo amb la segona prova de vol. En aquesta prova de vol ens trobem en un espai interior, on no és possible l'obtenció de senyal GPS. Repetim el procediment de l'apartat anterior. En aquest cas, al no obtenir senyal GPS el dron no sap en cap moment en la posició en la qual es troba i per tant no pot mantenir-se quiet. Com veiem en el vídeo el dron no fixa la posició en cap, segueix movent-se de forma lleugera.

Enllaç al vídeo: [https://youtu.be/icr3GNxpV\\_c](https://youtu.be/icr3GNxpV_c)

### 3.3. Comprovació sistema de navegació indoor

En aquest apartat es troba la primera prova de vol on s'utilitza el sistema de navegació indoor dissenyat. Aquesta prova de vol es basa en la repetició de les proves de vol vist en els apartats anteriors però utilitzant el sistema de navegació indoor dissenyat. Com es pot veure en el vídeo, un cop enlairem el dron i activem el sistema, aquest fa que el dron mantingui la posició d'una forma molt bona.

Enllaç al vídeo: <https://youtu.be/mkMc0-XbPvs>

### 3.4. Navegació amb el sistema de navegació indoor

Aquest últim apartat corresponent als vídeos de les proves de vol correspon a la prova de vol on s'utilitza el sistema de navegació indoor dissenyat per navegar dins de l'espai que es troba el dron amb un parell d'ordres senzilles. En el vídeo s'observa com el dron entén les ordres que li donem a través del Mission Planner i les executa de forma suau i en tot moment equilibrat. Com es podrà veure en el vídeo, les ordres de vol que se li donen són a l'altura d'1 m. Durant el transcurs de la missió el dron compleix aquesta ordre.

Enllaç al vídeo: <https://youtu.be/D2DpwWztBUA>

## 4. Codi

### 4.1. Main

```
"""  
  
INDOOR NAVIGATION LOW COST SYSTEM FOR DRONES  
  
Developed by: Sergi Lucas Millan  
Email: sergilucasmillan@hotmail.com  
Tutor: Oscar Casas Piedrafita  
Center: UPC - EETAC  
  
File description: main code  
  
"""  
  
from RealsenseFunctions import *  
from ConnectionDroneFunctions import *  
import numpy as np  
import transformations as tf  
from dronekit import *  
  
# Configure cameras and establish connections  
pipeline_T265 = init_T265()  
pipeline_D435 = init_D435()  
  
vehicle = None
```

```

while vehicle is None:
    # Start connection with flight controller
    vehicle = droneConnection()

xdata = []
ydata = []
zdata = []

try:
    while True:

        print()

        # Start receiving images from both cameras
        frames_T265 = pipeline_T265.wait_for_frames()
        frames_D435 = pipeline_D435.wait_for_frames()

        # Get frames from cameras (trajectory and depth frames)
        pose_frame = frames_T265.get_pose_frame()
        depth_frame = frames_D435.get_depth_frame()

        # Get trajectory data from frame
        pose_data = pose_frame.get_pose_data()

        # Create quaternion matrix about vector rotation [wxyz] from
        # trajectory data (with correct reference frame)
        matrix_quaternion =
        tf.quaternion_matrix([pose_data.rotation.w, -pose_data.rotation.z,
        pose_data.rotation.x,
        -
        pose_data.rotation.y])

        # Get TaitBryan angles in rad from quaternion matrix
        TaitBryan_rad =
        np.array(tf.euler_from_matrix(matrix_quaternion, 'sxyz'))

        # Execute function to calculate distance to object
        distance_object =
        get_distance_pixels_inside_region(frames_D435, 1, 0, 0, 640, 480)

        # Get UNIX time
        current_time = int(round(time.time() * 1000000))

        # Rotation to correct reference frame
        x = -pose_data.translation.z
        y = pose_data.translation.x
        z = -pose_data.translation.y

        # Send message to flight controller about position and
        # rotation
        message_vision_position_estimate(vehicle, current_time, x, y,
        z, TaitBryan_rad[0], TaitBryan_rad[1],
        TaitBryan_rad[2])

        # Send message to flight controller of distance to object
        message_distance_sensor(vehicle, current_time, 20, 500,
        distance_object)

        # Save trajectory data in arrays (Save to excel, change to our

```

```

reference frame)
    xdata.append(y)
    ydata.append(x)
    zdata.append(z)

    time.sleep(1.0 / 30)

finally:

    # Close connections
    pipeline_T265.stop()
    print("LOG: T265 - Save video record as
{}_T265.bag".format(datetime.datetime.now().strftime("%Y%m%d_%H%M%S")))
    )

    pipeline_D435.stop()
    print("LOG: D435 - Save video record as
{}_D435.bag".format(datetime.datetime.now().strftime("%Y%m%d_%H%M%S")))
    )

    vehicle.close()
    print("LOG: Connections closed")

    # Save trajectory data to excel file
    save_to_excel(xdata, ydata, zdata)

```

## 4.2. Funcions relacionades amb les càmeres

```

"""
INDOOR NAVIGATION LOW COST SYSTEM FOR DRONES

Developed by: Sergi Lucas Millan
Email: sergilucasmillan@hotmail.com
Tutor: Oscar Casas Piedrafita
Center: UPC - EETAC

File description: includes functions where cameras Intel RealSense are
involved

"""

import pyrealsense2 as rs
import numpy as np
import datetime
import openpyxl as xl

# Configure T265 camera to stream pose data and vision from fisheye,
start recording and establish connection.
def init_T265():
    pipeline_T265 = rs.pipeline()
    config_T265 = rs.config()
    config_T265.enable_device('905312110153') # Detect T265 camera by
device serial
    config_T265.enable_stream(rs.stream.pose)

```



```

config_T265.enable_stream(rs.stream.fisheye, 1)
config_T265.enable_stream(rs.stream.fisheye, 2)

# Start record

config_T265.enable_record_to_file("{}_T265.bag".format(datetime.datetime.now().strftime("%Y%m%d_%H%M%S")))
pipeline_T265.start(config_T265)
print("LOG: T265 - Connected with camera")
print("LOG: T265 - Start recording")
return pipeline_T265

# Configure D435 camera to stream depth and color, start recording and establish connection
def init_D435():
    pipeline_D435 = rs.pipeline()
    config_D435 = rs.config()
    config_D435.enable_device('829212070982') # Detect D435 camera by device serial
    config_D435.enable_stream(rs.stream.depth, 848, 480,
rs.format.z16, 30)
    config_D435.enable_stream(rs.stream.color, 1280, 720,
rs.format.rgb8, 15)

    # Start record

config_D435.enable_record_to_file("{}_D435.bag".format(datetime.datetime.now().strftime("%Y%m%d_%H%M%S")))

    pipeline_D435.start(config_D435)
    print("LOG: D435 - Connected with camera")
    print("LOG: D435 - Start recording")
    return pipeline_D435

# Counts number of pixels with distance below than minimum distance (security distance) and then it calculates if is an obstacle or not
def get_distance_pixels_inside_region(frames_D435, minimum_distance, x_up, y_up, x_down, y_down):
    depth_frames = frames_D435.get_depth_frame()
    list_distances = []
    j = y_up

    # Iterate matrix to get all pixel distances
    while (j < y_down):
        i = x_up
        while (i < x_down):
            pixel_distance = depth_frames.get_distance(i, j)
            if (pixel_distance < minimum_distance and pixel_distance
!= 0):
                list_distances.append(pixel_distance)
            i = i + 1
        j = j + 1

    # Creates histogram from all pixel distance
    hist, bins = np.histogram(list_distances, bins='auto')

    index = np.argmax(hist)

    # Observes if number of pixels below security distance is greater than fixed value (4000 in this case)

```

```

if len(list_distances) > 4000:
    return bins[index]*100
else:
    return 1000

# Get distance to object (if no object is detected it returns 1
meter)
mean_distance = bins[index]

return mean_distance

# Save trajectory data to excel file. One file for day, and one
worksheet per flight in day. (In one day it only creates
# one file but it can create many worksheets as flights in the day).
def save_to_excel(xdata,ydata,zdata):
    wb_name =
    "{}.xlsx".format(datetime.datetime.now().strftime("%Y%m%d")) # The
file name is the day in format YYYYMMDD
    worksheet_name = str(datetime.datetime.now().strftime("%H%M%S")) #
The worksheet name is the time in format hhmmss
    wb = None
    # First tries to find if there is a file created for the day. If
exists, use this file. Else create a new file.
    try:
        wb = xlsw.load_workbook(wb_name)
        print("LOG: Created new worksheet in {}".format(wb_name))
    except:
        print("LOG: Create new .xlsx file")

    if wb is None:
        wb = xlsw.Workbook()

    worksheet = wb.create_sheet(worksheet_name) # Add worksheet to
excel file

    j = 1
    # Write trajectory data on the file
    while j <= len(xdata):
        worksheet.cell(row=j, column=1).value = xdata[j-1]
        worksheet.cell(row=j, column=2).value = ydata[j-1]
        worksheet.cell(row=j, column=3).value = zdata[j-1]
        j = j + 1

    wb.save(wb_name) # Save the file
    print("LOG: Saved trajectory data in {}".format(wb_name))

```

### 4.3. Funcions relacionades amb el dron

```

"""
INDOOR NAVIGATION LOW COST SYSTEM FOR DRONES

Developed by: Sergi Lucas Millan
Email: sergilucasmillan@hotmail.com
Tutor: Oscar Casas Piedrafita
Center: UPC - EETAC

File description: includes functions where drone flight controller

```

```
(Pixhawk 2.4.8) is involved

"""

from pymavlink import mavutil
from dronekit import connect

# Establish connection with flight controller
def droneConnection():
    connection_string = "/dev/serial=" # Select direction where is
the flight controller (UART ports)
    connection_baudrate = 921600 # Select connection baud rate
    print("LOG: Connecting with Drone...")
    vehicle = None

    try:
        vehicle = connect(connection_string, wait_ready=False,
baud=connection_baudrate)
        print("LOG: Connected with Drone")
        return vehicle

    except:
        return vehicle

# Send messages with position and orientation to flight controller
def message_vision_position_estimate(vehicle, time, x, y, z, roll,
pitch, yaw):
    message =
vehicle.message_factory.vision_position_estimate_encode(time, x, y, z,
roll, pitch, yaw)
    vehicle.send_mavlink(message)
    vehicle.flush()

# Send messages with distance to object
def message_distance_sensor(vehicle, time, min_distance, max_distance,
current_distance):

    type = 4 # For unknown distance sensor
    id = 0 # No idea what means

    message = vehicle.message_factory.distance_sensor_encode(0,
min_distance, max_distance, current_distance, type,
id,
mavutil.mavlink.MAV_SENSOR_ROTATION_NONE, covariance=0)
    vehicle.send_mavlink(message)
    vehicle.flush()
```